

UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO

CARRERA:
INGENIERÍA ELECTRÓNICA

Trabajo de titulación previo a la obtención del título de:
INGENIERA ELECTRÓNICA E INGENIERO ELECTRÓNICO

TEMA:
**AUTOMATIZACIÓN DE UNA FRESADORA CNC PARA EL TALLADO DE
PIEZAS DE MADERA EN LA EMPRESA POST-FORMADOS DEL VALLE**

AUTORES:
YADIRA MARIBEL PILATUÑA CARCHI
ISMAEL RAMIRO PILLAJO TIPANTUÑA

TUTOR:
JOSÉ ALCIDES RUMIPAMBA LÓPEZ

Quito, julio del 2018

CESIÓN DE DERECHOS DE AUTOR

Nosotros Yadira Maribel Pilatuña Carchi con documento de identificación N° 1721083051 e Ismael Ramiro Pillajo Tipantuña con documento de identificación N° 1721582482, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: “AUTOMATIZACIÓN DE UNA FRESADORA CNC PARA EL TALLADO DE PIEZAS DE MADERA EN LA EMPRESA POST-FORMADOS DEL VALLE”, mismo que ha sido desarrollado para optar por el título de Ingeniera Electrónica e Ingeniero Electrónico, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



Yadira Maribel Pilatuña Carchi
C.I. 1721083051



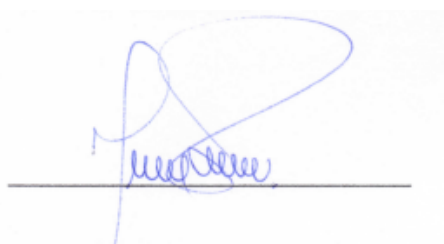
Ismael Ramiro Pillajo Tipantuña
C.I. 1721582482

Quito, julio del 2018

DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación, “AUTOMATIZACIÓN DE UNA FRESADORA CNC PARA EL TALLADO DE PIEZAS DE MADERA EN LA EMPRESA POST-FORMADOS DEL VALLE” realizado por Yadira Maribel Pilatuña Carchi e Ismael Ramiro Pillajo Tipantuña, obteniendo un producto que cumple con todos los requerimientos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, julio del 2018



José Alcides Rumipamba López

C.I: 1803854353

DEDICATORIA

Dedico este proyecto, primeramente, a Dios quien supo guiar mi camino, dándome la fortaleza para seguir adelante y no desmayar ante cada obstáculo que se me presentaba.

A mi padre Luis Pilatuña quien con su carácter supo sembrar en mi la responsabilidad de alcanzar una meta, por hacer de mi la persona que soy, inculcándome valores y principios para defenderme en el camino que me espera como profesional.

A mi madre Eulalia Carchi por su apoyo incondicional, por ser el motor de todos los días, mi compañera y amiga en cada momento difícil, brindándome su amor y comprensión.

A mis hermanas por estar siempre presentes en este camino llenándome de alegría cada uno de estos días. A Juan por ser parte de mi vida, mi apoyo en las buenas y malas, dándome valor y fortaleza para conseguir mi sueño.

Yadira

A Dios por la fortaleza para seguir adelante con mis objetivos, también a mi madre Fanny por guiarme por el buen camino y ayudarme en todo momento, a mi padre Ramiro por ayudarme con los estudios y ser mi ejemplo a seguir en la vida y profesionalmente, a mis hermanos por apoyarme en este largo camino.

A Adriana quien siempre ha estado a mi lado durante esta carrera y siempre me ha dado su apoyo incondicional y la fortaleza para no rendirme, a mi hija Andrea que con su amor es mi motivación personal para poder ser mejor cada día y por último a mis amigos por cada momento compartido en el transcurso de mi carrera universitaria.

Ismael

AGRADECIMIENTOS

Agradezco a Dios por llenar de bendiciones mi camino estudiantil, por escuchar mis oraciones y llenarme de sabiduría.

Agradezco en especial a la Universidad Politécnica Salesiana por haberme formado como profesional, enseñándome el camino del respeto y el amor a la profesión. A su vez a cada uno de los docentes que impartieron sus conocimientos orientándome con su sabiduría y enseñanza en el transcurso de mi formación académica.

Al ingeniero José Rumipamba por su valiosa colaboración y asesoramiento en la dirección de la presente tesis, mostrando interés en cada uno de sus tutorías, orientándonos de manera oportuna.

Al señor Thomas Lara por acogernos en su taller, apoyándonos con sus conocimientos, por la motivación que nos brindaba cada día, por no dejarnos solos, y convertirse en la mano derecha de esta tesis.

Gracias Isma por tu colaboración en esta tesis, por hacer de este un trabajo menos complicado, gracias por tu capacidad de esfuerzo y tu simpatía. A todos mis amigas y amigos de la Carrera, por la amistad que me brindaron, por el apoyo incondicional en momentos difíciles, por cada día de risas y estudio, los voy a extrañar.

Yadira Maribel Pilatuña Carchi

Primeramente, a Dios por haberme guiado en cada oportunidad que me da cada día de mi vida y por el largo camino que estuvo junto a mí.

Agradezco a la Universidad Politécnica Salesiana que me ha permitido estudiar la Carrera de Ingeniería Electrónica, también por la formación humana que he recibido durante la carrera.

Adicionalmente a los docentes quienes compartieron sus conocimientos durante mi formación y así adquirir un pensamiento profesional.

Quiero realizar un agradecimiento especial al ingeniero José Rumipamba por su calidad humana y profesional, quien nos apoyó de manera incondicional con su tiempo y conocimiento en el transcurso de este proyecto.

También quiero agradecer a Thomas Lara quien nos acogió en su taller, además nos ayudó y guio en la realización del proyecto, de igual manera a Alcides Andrade que de igual manera nos ayudó para culminar este objetivo.

Ismael Ramiro Pillajo Tipantuña

ÍNDICE GENERAL

CESIÓN DE DERECHOS DE AUTOR.....	i
DECLARATORIA DE COAUTORÍA DEL DOCENTE TUTOR	ii
DEDICATORIA	iii
AGRADECIMIENTOS	iv
ÍNDICE GENERAL	vi
ÍNDICE DE FIGURAS	ix
ÍNDICE DE TABLAS	xi
RESUMEN	xii
ABSTRACT	xiii
INTRODUCCIÓN	xiv
CAPÍTULO 1	1
ANTECEDENTES	1
1.1. Planteamiento del problema	1
1.2. Justificación.....	1
1.3. Objetivos	1
1.3.1. Objetivo General	1
1.3.2. Objetivo específicos.....	2
1.4. Alcance.....	2
CAPÍTULO 2	3
MARCO CONCEPTUAL	3
2.1. Control numérico por computadora	3
2.2. Principio de funcionamiento del control numérico en una fresadora	3
2.3. Tipos de fresas	4
2.4. Motor paso a paso	4

2.5. Software de control Matlab	5
2.6. Software Inkscape.....	5
2.7. Código G	6
2.8. Ejemplos de proyectos	6
2.8.1. Ejemplo 1	6
2.8.2. Ejemplo 2	7
2.8.3. Ejemplo 3	7
CAPÍTULO 3	8
DISEÑO E IMPLEMENTACIÓN	8
3.1. Arquitectura del sistema anterior de control de la empresa.....	8
3.2. Sistema propuesto para la automatización de la fresadora	10
3.3.1. Computador.....	10
3.3.2. Microcontrolador	10
3.3.3. Drivers	11
3.3.4. Actuadores	12
3.4. Desarrollo del software y adquisición de imagen.	14
3.4.1 Instalación de la librería GRBL en el Arduino	15
3.4.1.1. Configuración de la librería GRBL en el Arduino	17
3.4.1.2. Cálculo del número de pasos que debe ejecutar el motor	18
3.4.1.3. Parámetros de la librería GRBL	19
3.4.2. Creación de la GUI en Matlab	22
3.5. Vectorización de Imagen <i>jpg</i>	31
CAPÍTULO 4	38
PRUEBAS Y RESULTADOS.....	38
4.1. Prueba de tiempo de tallado de pieza en madera	38
4.2. Prueba de tiempo de cortado de pieza en madera	40
4.3. Discusión de resultados	42

4.3.1. Resultados obtenidos de la pregunta 1.....	43
4.3.2. Resultados obtenidos de la pregunta 2.....	45
4.3.3. Resultados obtenidos de la pregunta 3.....	47
4.3.4. Resultados obtenidos de la pregunta 4.....	50
CAPÍTULO 5	52
CONCLUSIONES Y RECOMENDACIONES	52
5.1. Conclusiones	52
5.2. Recomendaciones.....	53
REFERENCIAS	54
ANEXOS	

ÍNDICE DE FIGURAS

Figura 2.1 Nomenclatura de tipo de fresa	4
Figura 2.2 Motor Paso a Paso.....	4
Figura 2.3 Logo software de vectorización de imagen	6
Figura 3.1 Tipos de madera.....	8
Figura 3.2 Arquitectura del sistema anterior que disponía la empresa.	8
Figura 3.3 Plantilla y corte manual	9
Figura 3.4 Indumentaria de trabajo	9
Figura 3.5 Arquitectura para la automatización de la fresadora.....	10
Figura 3.6 Opción para incluir librerías dentro del software Arduino.....	15
Figura 3.7 Librería GRBL para Arduino.....	16
Figura 3.8 Programa GRBL	16
Figura 3.9 Monitor Serial con configuración GRBL.	17
Figura 3.10 Monitor serial del software Arduino.	17
Figura 3.11 Parámetros de configuración GRBL	19
Figura 3.12 Ventana de comandos de Matlab.	22
Figura 3.13 Ventana para creación de una nueva GUIDE.	22
Figura 3.14 GUI en blanco para el desarrollo de la interfaz.	23
Figura 3.15 Archivo <i>.m</i>	25
Figura 3.16 Archivo <i>.fig</i>	25
Figura 3.17 Código de comunicación entre Matlab y el Arduino.	25
Figura 3.18 Interfaz de comunicación entre Matlab y el Arduino.....	26
Figura 3.19 Ventana de cierre de puerto.	28
Figura 3.20 Código <i>CloseRequestFcn</i>	28
Figura 3.21 Código Pushbutton.	29
Figura 3.22 Configuración de un axes.	30
Figura 3.23 Código Axes	30
Figura 3.24 Ventana software Inkscape.	31
Figura 3.25 Carpeta imágenes <i>.jpg</i> y ventana <i>Inkscape</i>	32
Figura 3.26 Imagen seleccionada.	32
Figura 3.27 Ventana <i>Trace Bitmap</i>	33
Figura 3.28 Ventana <i>Brightness cutoff</i>	33

Figura 3.29 Imagen vectorizada.	34
Figura 3.30 Pestaña <i>Extensions</i>	34
Figura 3.31 Ventana <i>Tools library</i>	35
Figura 3.32 Especificaciones técnica del fresado.	35
Figura 3.33 Ventana <i>Path to Gcode</i>	36
Figura 3.34 Imagen vectorizada.	37
Figura 3.35 Vectorización de imagen a color.	37
Figura 4.1 Tallado de madera con la fresadora CNC.	38
Figura 4.2 Gráfica comparativa de tiempos	40
Figura 4.3 Corte de madera con la fresadora CNC.	41
Figura 4.4 Gráfica comparativa de tiempos de corte.	42
Figura 4.5 Resultados obtenidos del proceso manual.	43
Figura 4.6 Restos de polvo obtenidos de tallado.	44
Figura 4.7 Resultados obtenidos con el proceso automatizado.	44
Figura 4.8 Aspiradora para eliminar restos de polvo y aserrín.	45
Figura 4.9 Resultados obtenidos del proceso manual.	45
Figura 4.10 Tallado de pieza de madera realizada manualmente.	46
Figura 4.11 Resultados obtenidos del proceso con el sistema automatizado.	46
Figura 4.12 Tallado de pieza de madera realizada con el sistema automatizado.	47
Figura 4.13 Tallado de pieza de madera de alta dificultad	47
Figura 4.14 Resultados obtenidos del proceso manual.	48
Figura 4.15 Máquina manual con un motor grande.	48
Figura 4.16 Resultados obtenidos del proceso con el sistema automatizado.	49
Figura 4.17 Fresadora CNC con un motor pequeño.	49
Figura 4.18 Resultados obtenidos del proceso manual.	50
Figura 4.19 Operario realizados el corte manualmente.	50
Figura 4.20 Resultados obtenidos del proceso con el sistema automatizado.	51
Figura 4.21 Operario realizando el corte o tallado desde el computador.	51

ÍNDICE DE TABLAS

Tabla 3.1 Especificaciones Técnicas del Arduino Uno	11
Tabla 3.2 Especificaciones técnicas del driver HY-DIV268N-5A.....	12
Tabla 3.3 Especificaciones técnicas del motor paso a paso	14
Tabla 3.4 Requerimiento para el control de motores paso a paso	14
Tabla 3.5 Configuración de los sensores de final de carrera.....	21
Tabla 3.6 Herramientas interfaz GUI.....	23
Tabla 3.7 Componentes interfaz GUI.	24
Tabla 4.1 Tiempos de tallado con sistema de automatización.	39
Tabla 4.2 Tiempo de tallado realizado por el operario	39
Tabla 4.3 Tiempos de pruebas de corte en piezas de madera	41
Tabla 4.4 Tiempo de corte realizado por el operario.	41
Tabla 4.5 Preguntas del funcionamiento de la Fresadora CNC.....	43

RESUMEN

La Empresa Post-formados del Valle durante quince años se ha dedicado a la fabricación de muebles de cocina, oficina y puertas de madera, para la ejecución de los tallados y cortes la empresa dispone de una fresadora manual, en la que el operario debe dar forma al diseño que se va a modelar en dichos muebles, razón por la cual, en el presente trabajo se realizó el proceso de automatización de una fresadora CNC, cuyo funcionamiento es controlado mediante una computadora con Matlab, código G y una tarjeta Arduino Uno. Este proceso permitió a la empresa de tallado de madera resolver problemas de producción en el diseño de muebles. Se utilizó motores paso a paso para proporcionar movimientos precisos, estos son controlados mediante un software MATLAB que interpreta el Código G utilizado en control numérico. Para intercomunicar el software MATLAB y los motores se empleó un controlador Arduino para leer el código G y enviar señales a los motores indicándoles los movimientos que debe hacer. Una vez implementado el sistema automático en la fresadora, se realizó las pruebas respectivas que mostraron de forma gráfica el buen desempeño de dicho sistema y las ventajas en seguridad y tiempo que la empresa obtuvo del mismo.

Palabras claves: Arduino, CNC (control numérico por computadora), Código G, Fresadora, Matlab.

ABSTRACT

The company “Post-Formados del Valle” during fifteen years has been dedicated to the manufacture of furniture of kitchen, office and doors of wood, for the execution of the carvings and cuts of the company has of a manual of milling machine, and of which the operator must shape the design that will be modeled on said furniture, which is why, in the present work the automation process of a CNC milling machine was carried out, whose operation is controlled by a computer with Matlab, G code and an Arduino Uno card. This process allowed the wood carving company to solve production problems in the design of furniture. Stepper motors used to provide precise movements; a MATLAB software that interprets the G code used in numerical control controls these. To intercommunicate the MATLAB software and the motors, an Arduino controller used to read the G code and send signals to the motors indicating the movements that must be done. Once the automatic system implemented in the milling machine, the respective tests performed, which graphically showed the good performance of said system and the advantages in safety and time that the company obtained from it.

Keywords: Arduino, CNC (numerical control by computer), G code, Milling machine, Matlab.

INTRODUCCIÓN

La Empresa Post-Formados del Valle está ubicada en Amagüña, dicha empresa se dedica a la fabricación, tallado y corte de muebles de cocina, siendo su mercado principal la construcción domiciliaria. La madera utilizada como materia prima para la elaboración de los muebles es MDF.

El tallado o corte de cada pieza de madera es desarrollada de forma manual, siendo susceptible de pérdidas de materia prima, generando tiempos de trabajo extensos.

En el presente trabajo se exponen la automatización de una fresadora CNC, partiendo de conocimientos teóricos y conceptuales del funcionamiento de este tipo de máquinas.

El primer capítulo, expone el planteamiento del problema, su justificación y los objetivos tanto general como específicos a ser desarrollados.

En el segundo capítulo, se detalla los antecedentes que presenta la empresa y los materiales con los que trabaja. Se indica conceptos de carácter electrónicos, donde se exponen definiciones sobre lenguajes de programación y otros conceptos relacionados con el proyecto y se ponen énfasis en cuanto a los motores que se debe utilizar.

En el tercer capítulo, se expone la arquitectura anterior que presenta la empresa en el tallado de piezas y la mejora mediante la adquisición y automatización de la fresadora CNC. Se indica los cálculos de precisión y velocidad para seleccionar los motores y el driver de control. Contiene el desarrollo de la interfaz de control mediante el uso del software MATLAB y la vectorización de imágenes con el software Inkscape.

El cuarto capítulo, se muestra las pruebas realizadas de funcionamiento en el tallado y corte de piezas de madera realizadas con el sistema implementado. Estas pruebas fueron valoradas mediante encuestas que determinan el buen desempeño de la máquina.

Finalmente, el quinto capítulo, expone las conclusiones y recomendaciones encontradas a lo largo del desarrollo de este trabajo.

CAPÍTULO 1

ANTECEDENTES

1.1. Planteamiento del problema

Las prácticas actuales que se realizan en la empresa Post-formados del Valle, para la elaboración de piezas de madera está limitada debido a que es necesario la presencia del operario durante el proceso de tallado o corte, además el operario utiliza un molde que debe ser realizado con anterioridad, el mismo debe tener un corte de calidad para que la pieza final no presente fallas. El operario al estar en contacto con los residuos de madera es expuesto a daños físicos, ocasionando problemas respiratorios y lesiones físicas.

La empresa dispone de una fresadora manual, que trabaja en un solo eje, haciendo que el operario tenga que manipular la pieza para obtener el tallado solicitado, los procesos de acabado son complejos de realizar, porque la manipulación de la pieza debe ser precisa.

Por otra parte, la calidad y el tiempo de trabajo no son óptimos, ya que existe desperdicio de materia prima en el proceso de tallado.

1.2. Justificación

La empresa requiere la automatización de una máquina fresadora CNC para el tallado de piezas de madera, la misma que proporcionará beneficios a la empresa dotándole de ventajas en tiempo y seguridad en el proceso de producción.

Mediante la operación de esta máquina, la empresa logrará realizar tallados que en la actualidad tiene una alta complejidad, ya que dispone de una fresadora manual que ejecuta su trabajo en un solo eje en grandes lapsos de tiempo y sin las seguridades necesarias.

1.3. Objetivos

1.3.1. Objetivo General

Desarrollar la automatización de una Fresadora CNC para el tallado de piezas de madera de alta complejidad en la empresa Post-formados del Valle.

1.3.2. Objetivo específicos

- Caracterizar el funcionamiento de una fresadora CNC para identificar las variables a controlar.
- Diseñar el sistema electrónico de la fresadora CNC para el control de la máquina a través de una computadora.
- Desarrollar un software de control y su interfaz que permita la automatización de la fresadora para el tallado de pieza de madera.
- Implementar los sistemas diseñados para el mejoramiento de la fresadora.
- Validar mediante pruebas y resultados el funcionamiento de la fresadora CNC, en piezas de diferente tipo de madera utilizada en la empresa.

1.4. Alcance

El presente proyecto brindará una solución en cuanto al diseño y producción de tallado de piezas de madera. Esto se consigue aplicando un sistema de automatización, mediante el uso de sistemas electrónicos, que permitan reducir los tiempos en el proceso de producción y, además, del ahorro de materia prima al momento del tallado.

La fresadora dispone de 3 ejes que están representados por tres motores a pasos. Para el tallado de las piezas se dispondrá de un kit de fresas, que se utilizará para los diferentes diseños. Estas fresas serán sustituidas de forma manual.

El usuario seleccionará el patrón mediante un HMI, especificando el modelo de tallado y el tipo de fresa que se utilizará. El computador enviará las coordenadas para el tallado de la pieza y la máquina automáticamente realizará el diseño solicitado.

CAPÍTULO 2

MARCO CONCEPTUAL

2.1. Control numérico por computadora

El control numérico por computador realiza varios procesos, que implican dar movimiento velocidad y posición a un dispositivo mecánico mediante órdenes enviadas en tiempo real.

Estos procesos se dividen en dos categorías: el proceso herramienta-máquina como por ejemplo tenemos el taladro, fresadora, laminado o torneado; y el proceso sin herramienta-máquina como por ejemplo el ensamblaje, trazado, oxicorte o metrología.

Para cada uno de estos procesos interviene la utilización de un sistema de coordenadas que describe el movimiento del husillo, dando como resultado el diseño o ensamblado de piezas. (Teruel, 2005)

2.2. Principio de funcionamiento del control numérico en una fresadora

El control numérico ha generado un avance para las aplicaciones de las máquinas industriales mediante la automatización de las mismas, este control permite a las fresadoras realizar movimientos que para el operario son difíciles de ejecutar manualmente, como por ejemplo círculos o figuras con líneas curvas.

La automatización de la máquina optimiza algunas variables que en la empresa son muy importantes para generar utilidades como son: productividad, tiempo de producción, precisión, seguridad y reducción de la pérdida de materia prima.

Este sistema basado en el control numérico envía desde un computador instrucciones hacia un controlador para generar el movimiento de los motores.

Los movimientos que realiza la fresadora son los siguientes:

- Movimiento eje Z.- Donde se encuentra la herramienta cortante con la fresa.
- Movimiento eje X.-Realiza el movimiento horizontal de la mesa.
- Movimiento eje Y.- Realiza el movimiento vertical de la mesa. (Hernandez, 2012).

2.3. Tipos de fresas

Las fresadoras utilizan diferentes tipos de husillos que se denominan fresas, estas fresas son herramientas de corte que permiten dar forma o diseño a distintos materiales.

Figura 2.1 Nomenclatura de tipo de fresa

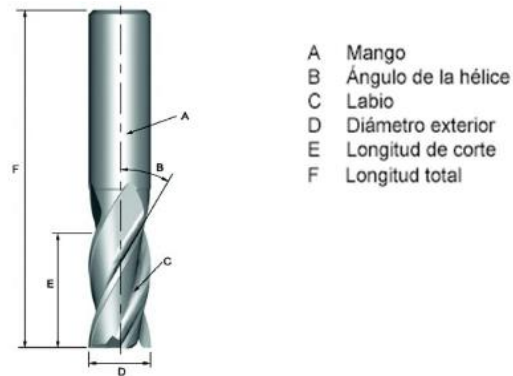


Diagrama de fresadora circular (Saavedra, 2015)

Las fresas pueden clasificarse según su mecanismo de tallado o arranque de viruta, esto se realiza con la ayuda de movimientos rotativos que realiza la máquina. (Saavedra, 2015)

2.4. Motor paso a paso

Un motor paso a paso permite transformar energía eléctrica en mecánica, esta clase de motores gira en determinados ángulos lo que le permite realizar desplazamientos muy precisos.

Figura 2.2 Motor Paso a Paso



Motor Paso a Paso Nema 25 (Bricogeek, 2013)

Los motores paso a paso controlan el cambio de dirección de flujo de corriente a través de las bobinas que lo forman:

- Controlar el desplazamiento del rotor en función de las tensiones que se aplican a las bobinas, para obtener un desplazamiento hacia adelante y atrás.
- Controla el número de pasos por vuelta.
- Controla la velocidad del motor. (Carletti, 2007)

2.5. Software de control Matlab

Matlab es un software de uso estudiantil y profesional, además posee grandes prestaciones de desarrollo matemático con lenguaje propio el lenguaje M, además está disponible para varias plataformas tales como Unix, Mac OS X, Linux y Windows.

Matlab resuelve varios problemas matemáticos que intervienen matrices, ecuaciones, implementación de algoritmos, entre otros problemas matemáticos. Adicionalmente Matlab cuenta con el desarrollo de interfaces de usuario (GUI) y de Simulink (plataforma de simulación multidominio).

Matlab también permite comunicarse con varios dispositivos mediante comunicación serial muy utilizado en varios dispositivos tales como Arduino, PIC, Labview, entre otros. Esto es útil pues permite tener una gran variedad de uso en la actualidad con herramientas tecnológicas que se van implementando. (The MathWorks, 2016)

2.6. Software Inkscape

Inkscape es un editor de gráficos vectoriales de código abierto, similar a programas como Adobe Illustrator, Corel Draw, Freehand o Xara X. Lo que lo hace único es que usa como formato nativo el Scalable Vector Graphics (SVG), un estándar abierto de W3C basado en XML. (Inkscape, 1999)

Normalmente se prefiere el diseño vectorial para la crear logotipos, de ilustraciones y de creaciones que requieran una gran escalabilidad, es decir que se puedan ampliar o reducir a cualquier tamaño. Inkscape es un programa usado tanto a nivel particular como por una gran variedad de empresas.

Figura 2.3 Logo software de vectorización de imagen



Software que permite vectorizar imágenes jpg. (Inkscape, 1999)

2.7.Código G

El G-Code es un lenguaje de programación que se usa para el control de máquinas CNC, además permite el control de los motores mediante sentencias que describen el movimiento de los motores según las coordenadas especificadas.

El código G se lo puede guardar mediante un formato de texto (.txt), esto permite que se pueda realizar cambios en el texto y así también guardarlos. Para modificar el texto es necesario tener un programa de editor de texto como wordpad, pero lo que más se usa es un software que se especializa en generar los códigos G, la mayoría de estos son software libre.

El código G describe el movimiento y las diferentes operaciones que la máquina CNC debe realizar para la fabricación del objeto que se desea plasmar. Actualmente el código G se ha convertido en una herramienta para el control numérico por computadora. (Smith, 2003)

2.8. Ejemplos de proyectos

2.8.1. Ejemplo 1

Tema: Automatización de una fresadora artesanal a control numérico mediante un ordenador utilizando software.

Este proyecto fue realizado por Roberto Washington Alava Navas y José Antonio Palacios Bauz en la Escuela Superior Politécnica del Litoral ubicada en Guayaquil-Ecuador en el año 2014.

En el proyecto se realiza la automatización de una fresadora torreta convencional, los elementos electrónicos que se utilizaron fueron los siguientes: motores paso a paso, drivers de control, controlador Breakout board, además utilizaron el software Enhanced Machine Control (EMC) para generar el código G de varios modelos y el software Mach3 para el control y calibración de la máquina fresadora CNC,

adicionalmente se usa un computador para la comunicación entre el software de control y el controlador. (Alva & Palacios, 2014)

2.8.2. Ejemplo 2

Tema: Diseño y Construcción de una fresadora con control numérico por computador para uso didáctico.

Este proyecto fue realizado por Oscar Alejandro Cruz García, Evelyn Rebeca Guerra López, Ricardo Alfredo Herrera Castro y Néstor Oswaldo Meza Orellana en la Universidad de El Salvador ubicada en San Salvador-El Salvador en el año 2014.

En el proyecto se realiza el diseño y la construcción de una fresadora CNC, los elementos electrónicos que se utilizaron fueron los siguientes: motores paso a paso, drivers de control, controlador Arduino Uno, además utilizaron el software Inkscape para generar el código G de varios modelos y el software TxapucCNC_TX para el control y calibración de la máquina fresadora CNC, adicionalmente se usa un computador para la comunicación entre el software de control y el controlador.

Además, se realiza el diseño mecánico de la máquina fresadora CNC como la estructura básica, componentes mecánicos de la máquina y otros. (Cruz, Guerra, Herrera, & Meza, 2014)

2.8.3. Ejemplo 3

Tema: Diseño e implementación de un sistema de control numérico por computadora de una fresadora de madera en tres dimensiones.

Este proyecto fue realizado por Miguel Ángel Silva Fur y Teófilo Choque Condori en la Universidad Nacional del Altiplano ubicada en Puno-Perú en el año 2017.

En el proyecto se realiza el diseño y la construcción de una fresadora CNC, los elementos electrónicos que se utilizaron fueron los siguientes: motores paso a paso, drivers de control y el controlador fueron diseñados por los autores del proyecto, además utilizaron el software Inkscape para generar el código G de varios modelos y el software Mach3 para el control y calibración de la máquina fresadora CNC, adicionalmente se usa un computador para la comunicación entre el software de control y el controlador. (Silva & Choque, 2017)

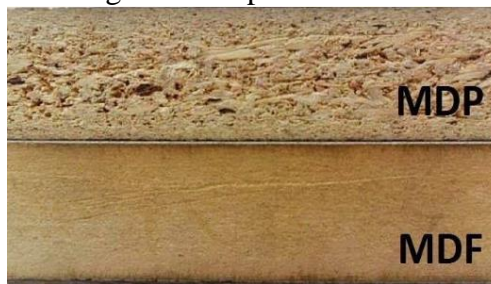
CAPÍTULO 3

DISEÑO E IMPLEMENTACIÓN

3.1. Arquitectura del sistema anterior de control de la empresa

La Empresa Post-formados del Valle está dedicada al diseño y fabricación de muebles de cocina, como también al tallado de puertas. Esta empresa está localizada en Amagüña, y viene operando 15 años con la supervisión de su propietario el Sr. Thomas Lara. Para la fabricación y tallado de piezas de madera, la empresa trabaja con materiales como MDF y MDP (aglomerado).

Figura 3.1 Tipos de madera



Madera MDF y MDP utilizadas en la empresa. (Do, 2014)

El sistema de control anterior que presentaba la empresa es totalmente manual ya que el operador debe encender la máquina y guiar la pieza de madera para crear figuras con ayuda de un molde.

Figura 3.2 Arquitectura del sistema anterior que disponía la empresa.



Sistema de control actual que presenta la empresa. Elaborado por: Pilatuña & Pillajo

El sistema permitía escoger el tipo de fresa (ya sea de corte o tallado) con la que se deseaba trabajar dependiendo del molde y la complejidad de la figura como se muestra en la Figura 3.3.

Figura 3.3 Plantilla y corte manual



Método de corte de piezas mediante el uso de plantillas. Elaborado por: Pilatuña & Pillajo

Si la empresa contaba con el molde el tiempo de tallado de la figura no se prolongaba demasiado, pero si se deseaba crear una figura distinta a la del molde se presentaban problemas, ya que se debían crear uno y se perdía tiempo en el diseño y la elaboración del mismo.

Por otro lado, debido a la interacción directa del operario con la máquina fresadora se dejaba de lado la seguridad industrial al momento del tallado, lo que conllevaba riesgos sobre la integridad física de dicha persona. El operario debía estar concentrado para realizar un corte o tallado porque de no hacerlo podía sufrir accidentes con la fresa.

Esta máquina al encenderse genera un ruido ensordecedor para el operario. Al tallar o cortar la pieza de madera proporciona residuos de viruta o polvo, que para el operario genera problemas en el sistema respiratorio y de la visión. Para prevenir estos problemas el operario utiliza indumentaria de seguridad como, por ejemplo: guantes, audífonos, mascarilla y gafas.

Figura 3.4 Indumentaria de trabajo



Indumentaria utilizada por los operarios para el corte y tallado de piezas, polvo generado. Elaborado por: Pilatuña & Pillajo

3.2. Sistema propuesto para la automatización de la fresadora

Actualmente en la empresa cuenta con una maquina artesanal para el tallado de piezas de madera, la propuesta para mejorar el tallado de las piezas es la automatización de la fresadora, para lo cual la empresa se comprometió en realizar la estructura mecánica de la fresadora, y la automatización será realizada según lo previsto.



Sistema de control que permite la automatización de la fresadora. Elaborado por: Pilatuña & Pillajo

Para automatizar es necesario contar con los siguientes materiales: un computador, cuatro Motores paso a paso, un controlador con comunicación serial compatible con el computador, un driver por cada motor, una fuente de alimentación para los drivers de los motores.

3.3.1. Computador

Para la automatización de la máquina se dispuso de un computador en cual se desarrolla el software de control. Este software fue desarrollado dentro de un entorno de Matlab, permitiendo mediante una interfaz gráfica el control de los motores.

Los softwares necesarios para la automatización de la fresadora son:

- Matlab: permite el control de la máquina por programación.
- Inkscape: realiza la vectorización de las imágenes.

3.3.2. Microcontrolador

El microcontrolador que se utiliza para la automatización es el Arduino UNO, este permite realizar comunicación serial entre la computadora y el Arduino.

Para poder enlazar el Arduino con los drivers que controlan los motores paso a paso, se debe incluir la librería GRBL, esta librería contiene la información de un

controlador CNC. La información que brinda esta librería, está basada en la velocidad, aceleración de cada uno de los motores en sus respectivos ejes.

Para la adquisición de este microcontrolador se basó en las siguientes características técnica.

Tabla 3.1 Especificaciones Técnicas del Arduino Uno

Tarjeta	Arduino UNO
Microcontrolador	Atmega328
Voltaje de operación	5V
Voltaje entrada	Recomendada: 7-12V Límite: 6-20V
Pines entrada y salida digital	14 (6 PWM)
Pines de entrada analógica	6
Corriente continuo por pin IO	40mA
Memoria flash	32KB
SRAM	2KB
EEPROM	1KB
Frecuencia de reloj	16Mhz

Cuadro de especificaciones técnicas necesarias para el control de la fresadora.
(Gobierno de Canarias, 2013)

3.3.3. Drivers

Los drivers se escogerán según las especificaciones de los motores como la corriente y el voltaje con los que trabaja que se menciona en la siguiente sección.

Estos drivers deben permitir controlar los motores paso a paso de acuerdo a los micro pasos que se necesiten para desarrollar el tallado y corte de una pieza de madera.

De acuerdo a las especificaciones los drives a ser utilizados son: HY-DIV268N-5A ya que contienen un chip TB6600 que es un módulo, que permite obtener una alta precisión y estabilidad al momento de posicionar a cada uno de los motores.

Tabla 3.2 Especificaciones técnicas del driver HY-DIV268N-5A

Código	DRIVER HY-DIV268N-5A
Chip	TB6600
Compatible con motores	2 y 4 fases
Voltaje Entrada	8 - 50 V dc
Corriente Entrada	1 – 5A
Voltaje de trabajo	12 – 42 V
Corriente de salida	5.0 A (Valor nominal máximo absoluto, pico de 100ms) 4.5 A (Rango de operación, valor máximo)
Modo Selección Micro pasos	1/1, 1/2, 1/4 , 1/8, 1/16 pasos
Control de Corriente	0.2 – 5 A
Peso	200 g

Cuadro de especificaciones técnicas del driver a ser utilizado para el control de los motores (Electroniclub, s.f.)

3.3.4. Actuadores

Para justificar la utilización de los motores paso a paso, se tomó en cuenta la necesidad de que dichos motores deben ejercer pequeños movimientos, es decir se deben mover un paso a la vez. Cada motor debe ser colocado en cada uno de los ejes, permitiendo a la máquina el movimiento de acuerdo al tallado solicitado.

Para la selección de los motores paso a paso se tomó en cuenta los siguientes criterios.

- Se debe tener precisión en el tallado ya que se trabaja en milímetros.
- Se debe calcular la velocidad de trabajo de los motores, al igual que la de corte.
- La potencia de trabajo que permite mover la mesa.

Estos criterios permiten calcular el torque necesario para la adquisición de los motores. Para calcular el torque se utiliza la siguiente fórmula:

$$T = \frac{F \cdot L}{2 \cdot \pi \cdot E} \quad \text{Ec. (3.1)}$$

Donde:

F: Carga

L: Distancia recorrida por la tuerca en una revolución.

E: Eficiencia

Para obtener la carga se debe considerar lo siguiente:

- Peso de la mesa: 200lbs
- Peso de los tornillos sin fin: 15lbs

$$F = 200 + 15 = 215 \text{ lbf}$$

Ec. (3.2)

$$L = Px * Nw$$

Ec. (3.3)

Donde:

Px: Paso axial (in): 10mm(0.393701in)

Nw: Número de dientes: 1

$$L = 0.39370 * 1 = 0.39370 \text{ in}$$

$$E = (\tan \lambda) * \left[\frac{\cos \phi - f \tan \lambda}{\cos \phi \tan \lambda + f} \right]$$

Ec. (3.4)

Donde:

λ : ángulo de avance

ϕ : ángulo en el plano normal: 25°

f: coeficiente de fricción: 0.16

$$\tan \lambda = \left[\frac{L}{\pi * Dw} \right]$$

Ec. (3.5)

Donde:

Dw: es el diámetro: 0.704 in

$$\tan\lambda = \left[\frac{0.39370}{\pi*0.704} \right]$$

$$\tan\lambda = 0.178$$

$$E = (0.178) * \left[\frac{0.99-0.16+0.178}{0.99*0.178+0.16} \right]$$

$$E = 0.509$$

$$T = \frac{215*0.3937}{2*\pi*0.509}$$

$$T = 26.46\text{ lbf} - \text{in} = 423.84 \text{ onz} - \text{in}$$

$$T = 30.4\text{ Kg.cm}$$

Los cálculos realizados para la obtención del torque dan como resultado 30.4 Kg.cm, con este valor se determinó los motores a utilizarse para cada uno de los ejes. El valor del torque calculado se puede visualizar en la tabla 3.3.

Tabla 3.3 Especificaciones técnicas del motor paso a paso

Código	57BYGH115-003B
Dimensiones del marco	NEMA 23
Ángulo por paso	1.8 GRADOS
Voltaje	6.3 V
Corriente	3 A
Resistencia	2.1Ω
Inductancia	9mH
Torque	30 Kg.cm
Inercia del rotor	810 gcm ²
Número de alambres	4
Peso	1.55 Kg
Longitud	115 mm

Cuadro de especificaciones técnicas de motor paso a paso de acuerdo a cálculos realizados.
(Changzhou Wantai, 2009)

3.4. Desarrollo del software y adquisición de imagen.

Para el desarrollo de la interfaz se requirió del siguiente software:

Tabla 3.4 Requerimiento para el control de motores paso a paso

REQUERIMIENTOS	DESCRIPCIÓN
MATLAB	El uso del software Matlab permitirá realizar la interfaz y la comunicación serial con el Arduino.
InkScape	Este software permite vectorizar las imágenes a ser talladas, pasar de .jpg a .svg.
LIBRERÍA GRBL ARDUINO	Este firmware permite controlar los pulsos que se envían a cada driver de los motores.

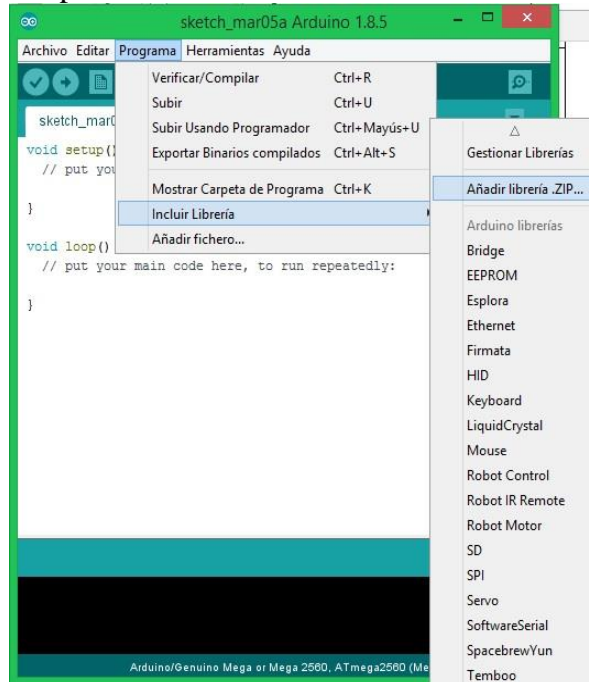
Tabla que indica los software y librería necesarios para el desarrollo de la interfaz que controla máquina. Elaborado por: Pilatuña & Pillajo

3.4.1 Instalación de la librería GRBL en el Arduino

Para instalar se descargó la librería que se encuentra en el siguiente link: <https://github.com/Protoneer/GRBL-Arduino-Library> , está en formato ZIP.

Se ingresó al software Arduino, Programa, incluir librería, añadir librería ZIP, abrir la librería descargada, como se muestra en la siguiente figura.

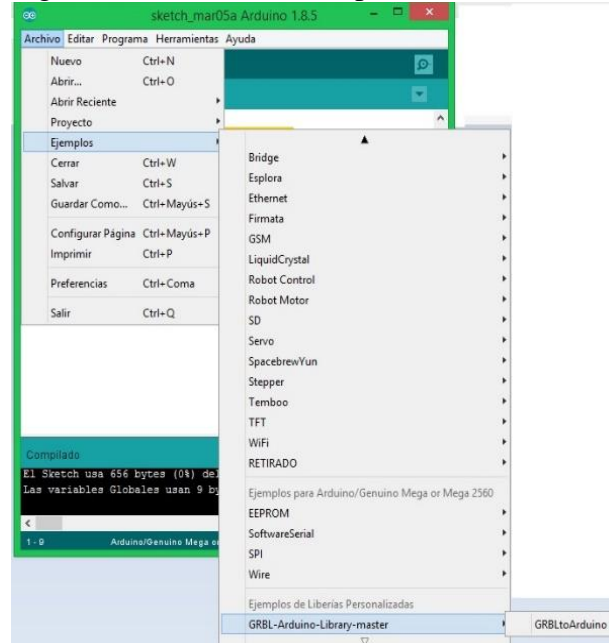
Figura 3.6 Opción para incluir librerías dentro del software Arduino.



Con esta opción se incluye la librería GRBL para el control de los motores. (Arduino, 2018)

Después de añadir la librería se buscó en Archivo, ejemplos, GRBL to Arduino.

Figura 3.7 Librería GRBL para Arduino



Librería GRBL para control de velocidad y aceleración de los motores. (Arduino, 2018)

Se cargó la librería al Arduino.

Figura 3.8 Programa GRBL

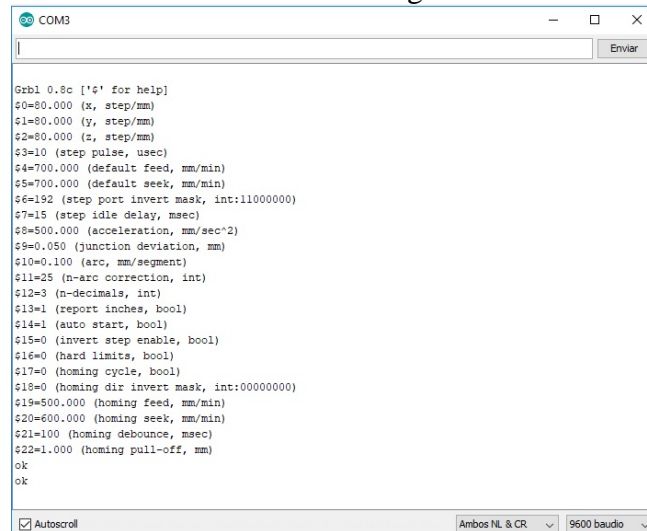


Programa con la configuración respectiva para el funcionamiento de los motores.(Arduino, 2018)

A continuación, se utilizó el monitor serial del Arduino para realizar la configuración de los motores, tanto en velocidad y aceleración de trabajo, así como también la velocidad de transmisión con el Matlab.

Para configurar se ingresó el símbolo \$\$\$. Por otra parte, para modificar cada parámetro se ingresó el símbolo \$ + el número de cada parámetro. Por ejemplo \$0=100.00 que representa los pasos en el eje x.

Figura 3.9 Monitor Serial con configuración GRBL.



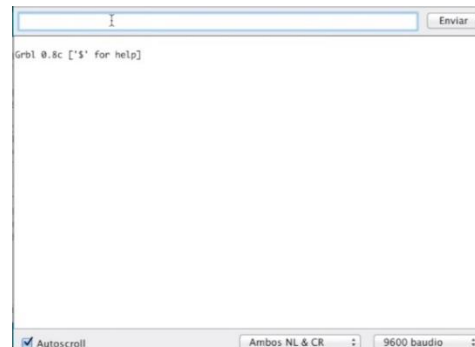
Contiene la configuración de velocidad y aceleración para cada uno de los motores. (Arduino, 2018)

3.4.1.1. Configuración de la librería GRBL en el Arduino

Para configurar la librería GRBL en el Arduino es necesario abrir el monitor serial y especificar cada uno de los parámetros, como por ejemplo la velocidad de comunicación de la tarjeta; que en este caso la velocidad es de 9600 baudio.

Al establecerse la comunicación con el Arduino se desplegará una línea de código como se muestra en la Figura 3.10.

Figura 3.10 Monitor serial del software Arduino.



Monitor serial del software Arduino permite modificar los parámetros del movimiento de los motores (Arduino, 2018)

Para conocer los parámetros a configurar se debe digitar el siguiente símbolo \$\$ en la barra en blanco del monitor serial, con este símbolo se puede observar cada uno de los parámetros necesarios en la maquina CNC.

Los parámetros se deben configurar digitando el signo \$ seguido del número de parámetro como por ejemplo \$4.

3.4.1.2. Cálculo del número de pasos que debe ejecutar el motor

Para conocer el número de pasos que el motor debe realizar en un milímetro se utilizó la siguiente ecuación.

$$pasos = \frac{360^\circ}{paso} \quad \text{Ec. (3.6)}$$

Donde:

360° = es el número de grados que da un motor en una vuelta.

Paso = tamaño del paso de un motor en grados.

Utilizando la ecuación 3.6 se encontró el número de pasos del motor paso a paso elegido para este trabajo y que se especifica en la sección 3.3.4. Conociendo el paso de motor que es 1.8 grados según las especificaciones del mismo se obtiene el siguiente resultado:

$$pasos = \frac{360^\circ}{1.8^\circ} = 200$$

Cuando se haya obtenido el número de pasos en una vuelta, se debe conocer el desplazamiento por vuelta del tornillo sin fin que se va a emplear en la construcción de la fresadora, para este caso será de 10 milímetros/paso.

$$\frac{200}{10 \text{ mm/paso}} = 20 \text{ paso/mm} \quad \text{Ec. (3.7)}$$

Luego se debe de multiplicar los micro paso configurados en el driver TB6600, para este caso será 4 micro pasos.

$$20 \times 4 = 80 \text{ paso/mm} \quad \text{Ec. (3.8)}$$

Con las operaciones realizadas se determinó el valor en milímetros que da un motor en una vuelta.

3.4.1.3. *Parámetros de la librería GRBL*

La librería GRBL presenta los siguientes parámetros para la configuración de cada uno de los motores en sus respectivos ejes.

Figura 3.11 Parámetros de configuración GRBL

```
Grbl 0.8c ['$' for help]
$0=80.000 (x, step/mm)
$1=80.000 (y, step/mm)
$2=80.000 (z, step/mm)
$3=10 (step pulse, usec)
$4=700.000 (default feed, mm/min)
$5=700.000 (default seek, mm/min)
$6=192 (step port invert mask, int:11000000)
$7=15 (step idle delay, msec)
$8=500.000 (acceleration, mm/sec^2)
$9=0.050 (junction deviation, mm)
$10=0.100 (arc, mm/segment)
$11=25 (n-arc correction, int)
$12=3 (n-decimals, int)
$13=1 (report inches, bool)
$14=1 (auto start, bool)
$15=0 (invert step enable, bool)
$16=0 (hard limits, bool)
$17=0 (homing cycle, bool)
$18=0 (homing dir invert mask, int:00000000)
$19=500.000 (homing feed, mm/min)
$20=600.000 (homing seek, mm/min)
$21=100 (homing debounce, msec)
$22=1.000 (homing pull-off, mm)
```

Librería GRBL del Arduino para la configuración de los motores. (Arduino, 2018)

\$0=80.000 (x, step/mm): Es el parámetro para configurar el número de pasos que realiza el motor en 1mm en el eje x de acuerdo a los resultados obtenidos en la sección 3.4.1.2.

\$1=80.000 (y, step/mm): Es el parámetro para configurar el número de pasos que realiza el motor en 1mm en el eje y de acuerdo a los resultados obtenidos en la sección 3.4.1.2.

\$2=80.000 (z, step/mm): Es el parámetro para configurar el número de pasos que realiza el motor en 1mm en el eje z de acuerdo a los resultados obtenidos en la sección 3.4.1.2.

\$3=10 (step pulse, usec): Este parámetro configura el ancho de pulso en microsegundos que reconoce el Arduino, este valor se lo deja por defecto, ya que así el Arduino no genera errores en el funcionamiento de los motores.

\$4=700.00 (default feed, mm/min): Este parámetro configura la velocidad de los motores con carga, esta velocidad puede ser menor dependiendo de las pruebas realizadas, o también puede ser igual al parámetro \$5.

\$5=700.00 (default seek, mm/min): Este parámetro configura la velocidad de los motores cuando busca una posición. El valor de la velocidad depende de las pruebas realizadas.

\$6=192 (step port invert mask, int: 11000000): Este parámetro invierte la dirección de los motores, este valor se lo deja por defecto.

\$7=15 (step idle delay, msec): este parámetro configura el tiempo de retardo después de realizar una orden. Este valor se lo deja por defecto ya que la maquina funciona bien.

\$8=500.000 (aceleración, $\frac{mm}{segundo^2}$): Este parámetro configura la aceleración que depende de la capacidad de los motores. Este valor se lo debe colocar mediante la realización de pruebas de aceleración al colocar valores en un rango de 30 a 1000 $\frac{mm}{segundo^2}$. El valor de la aceleración utilizado en este trabajo es de 500, el mismo que se seleccionó por inspección y que permitió al motor trabajar sin pérdidas de pasos.

\$9=0.050 (junction deviation, mm): Este parámetro configura el trazo de líneas curvas o los arcos. Este valor se lo deja por defecto.

\$10=0.100 (junction deviation, mm): Este parámetro configura el trazo de líneas curvas o los arcos. Este valor se lo deja por defecto.

\$11=25 (junction deviation, mm): Este parámetro configura el trazo de líneas curvas o los arcos. Este valor se lo deja por defecto.

\$12=3 (n-decimals, int): Este parámetro configura el número de decimales que aparecerán en los numero de la configuración.

\$13=1 (report inches, bool): Este parámetro configura las unidades en que trabajara la máquina. Para trabajar en milímetros se coloca 1, para trabajar en pulgadas se coloca 0.

\$14=1 (auto start, bool): Este parámetro configura el auto inicio, si se coloca el valor de cero se debe de programar un botón de inicio en el Arduino, entonces se debe colocar el valor de 1 para comenzar automáticamente.

\$15=0 (invert step enable, bool): Este parámetro permite invertir el pin de enable. Esto depende de la especificación del driver.

\$16=0 (hard limits, bool): Este parámetro se usa cuando se va a trabajar con finales de carrera para limitar el trabajo de la máquina. Para habilitarlo se debe de colocar el valor de 1.

\$17=0 (homing cycle, bool): Este parámetro se usa cuando está instalado los finales de carrera. Con esto la maquina se mueve hasta encontrar los finales de carrera.

\$18=0 (homing dir invert mask, int: 00000000): Este parámetro se usa para invertir la posición de los finales de carrera, ya que la librería supone que los finales de carrera están ubicados en los ejes positivos, y si están colocados en los ejes negativos se debe configurar según la siguiente tabla.

Tabla 3.5 Configuración de los sensores de final de carrera

Valor de Ajuste	Mascara	Eje X	Eje Y	Eje Z
0	00000000	No invertido	No invertido	No invertido
1	00000001	Invertido	No invertido	No invertido
2	00000010	No invertido	Invertido	No invertido
3	00000011	Invertido	Invertido	No invertido
4	00000100	No invertido	No invertido	Invertido
5	00000101	Invertido	No invertido	Invertido
6	00000110	No invertido	Invertido	Invertido
7	00000111	Invertido	Invertido	Invertido

Tabla de configuración de finales de carrera especificados en el ítem 18 de la configuración GRBL.

Elaborado por: Pilatuña & Pillajo

\$19=0 (homing feed, mm/min): Este parámetro configura la velocidad cuando regresa al origen con carga. Esta velocidad puede ser menor dependiendo de las pruebas realizadas, o también puede ser igual al parámetro \$20.

\$20=0 (homing seek, mm/min): Este parámetro configura la velocidad cuando regresa al origen sin carga. El valor de la velocidad depende de las pruebas realizadas.

\$21=100 (homing debounce, msec): Este parámetro configura el tiempo de espera en mili segundos cuando está regresando al punto de origen.

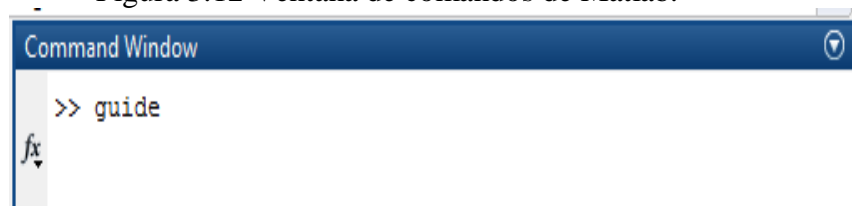
\$22=1.000 (homing pull-off, mm): Este parámetro configura el desplazamiento que realiza después de llegar al origen.

3.4.2. Creación de la GUI en Matlab

Para crear una GUI en Matlab se realizó los siguientes pasos:

- **Paso 1.** Ejecutar en el Matlab la siguiente instrucción en la ventana de comandos.

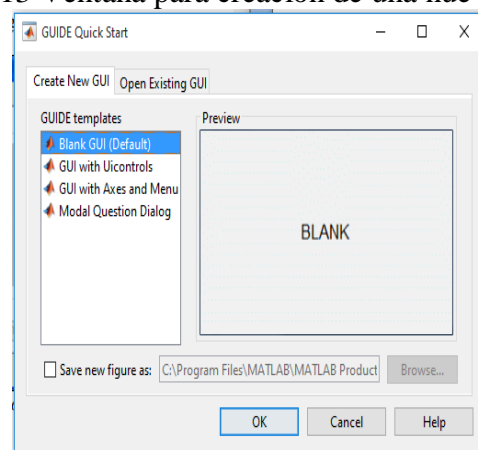
Figura 3.12 Ventana de comandos de Matlab.



Permite introducir órdenes directas. (The MathWorks, 2016)

Se abrirá el siguiente cuadro de dialogo, donde se encuentra las siguientes opciones:

Figura 3.13 Ventana para creación de una nueva GUIDE.



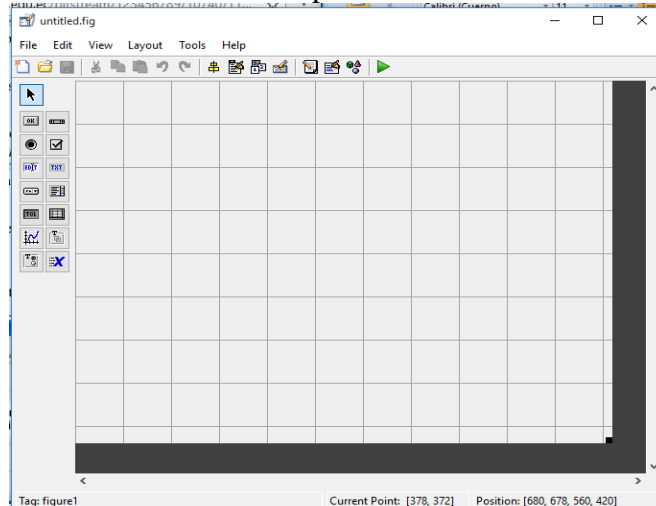
Creación de una nueva GUIDE. (The MathWorks, 2016)

- a) **Blank GUI(Default):** esta opción presenta una interfaz gráfica de usuario en blanco, en el cual se puede diseñar los programas.

- b) **GUI with Uicontrols:** esta opción presenta un ejemplo en cual se calcula la masa, la densidad y el volumen.
- c) **GUI with Axes and Menu:** esta opción es otro ejemplo el cual contiene un menú File con las opciones Open, Print y Close.
- d) **Modal Question Dialog:** con esta opción se muestra en la pantalla un cuadro de dialogo común, el cual consta de una pequeña imagen, una etiqueta y dos botones Yes y No, dependiendo del botón que se presione, el GUI retorna el texto seleccionado.

- **Paso 2.** De las opciones mencionadas seleccionamos **Blank GUI**.









Figura 3.14 GUI en blanco para el desarrollo de la interfaz.



Ventana para el desarrollo de la interfaz gráfica en Matlab. (The MathWorks, 2016)












La interfaz gráfica cuenta con las siguientes herramientas y componentes:

Tabla 3.6 Herramientas interfaz GUI

	Alinear objetos.
	Editor de menú.
	Editor de orden de etiqueta.
	Editor de herramientas
	Editor del M-File
	Propiedades de objetos
	Navegador de objetos
	Grabar y ejecutar

Herramientas que permiten configurar de manera rápida cada uno de los componentes de la interfaz.
(The MathWorks, 2016)

Tabla 3.7 Componentes interfaz GUI.

COMPONENTE	DESCRIPCIÓN
 Push Button	Invoca un evento inmediatamente.
 Slider	Usado para representar un rango de valores.
 Radio Button	Indica una opción que puede ser seleccionada.
 Check Box	Indica el estado de una opción o atributo.
 Edit Text	Caja para editar texto.
 Static Text	Muestra un string de texto en una caja.
 Pop-up Menu	Provee una lista de opciones.
 Listbox	Muestra una lista deslizable.
 Toggle Button	Solo dos estados on o off
 Panel	Agrupar botones como un grupo
 Button Group	Permite exclusividad de selección con los radio button.

Comandos para el desarrollo de la interfaz
(The MathWorks, 2016)

- **Paso 3.** Crear un archivo *.m* de nombre *inter_fin* que se despliega después de crear un archivo *.fig* del mismo nombre. El archivo *.m* es el que contiene el código que corresponde a los botones de control de la interfaz, y el archivo *.fig* contiene los elementos gráficos.

Figura 3.15 Archivo *.m*.

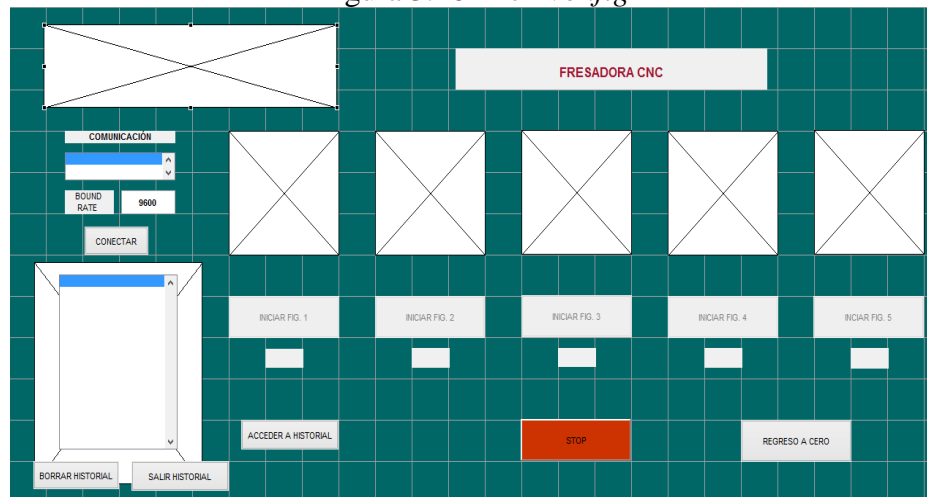
```

145 function conectar_Callback(hObject, eventdata, handles)
146 clc;
147 if strcmp(get(hObject,'String'),'CONECTAR')
148     serPortn = get(handles.puertos, 'Value');
149     if serPortn == 1
150         errordlg('Seleccione un puerto valido');
151     else
152         serList = get(handles.puertos,'String');
153         serPort = serList(serPortn);
154         serConn = serial(serPort, 'TimeOut', 1, ...
155             'BaudRate', str2num(get(handles.velocidad, 'String')));
156         try
157             fopen(serConn);
158             handles.serConn = serConn;
159             set(handles.fig_1,'enable','on');
160             set(handles.fig_2,'enable','on');
161             set(handles.fig_3,'enable','on');
162             set(handles.fig_4,'enable','on');
163             set(handles.fig_5,'enable','on');
164             set(hObject,'String','DESCONECTAR')
165         catch e
166             errordlg(e.message);
167         end
168     else
169         set(handles.fig_1,'enable','off');
170         set(handles.fig_2,'enable','off');
171         set(handles.fig_3,'enable','off');
172         set(handles.fig_4,'enable','off');
173         set(handles.fig_5,'enable','off');
174         set(handles.fig_5,'enable','off');
175         set(hObject, 'String','CONECTAR')
176         fclose(handles.serConn);
177     end
178     guidata(hObject, handles);

```

Archivo *.m* presenta el código de configuración de cada uno de los botones a utilizar en la interfaz. (The MathWorks, 2016)

Figura 3.16 Archivo *.fig*



Archivo *.fig*. interfaz con los componentes para el control de los motores. (The MathWorks, 2016)

- **Paso 4.** Configurar el puerto serial.

En la función de apertura de la interfaz se configura el siguiente código que permite la comunicación entre el Matlab y el Arduino.

Figura 3.17 Código de comunicación entre Matlab y el Arduino.

```

serialPorts = instrhwin('serial');
nPorts = length(serialPorts.SerialPorts);
set(handles.portlist, 'String', ...
    [{'Seleccione un Puerto'} ; serialPorts.SerialPorts ]);
set(handles.portlist, 'Value', 2);

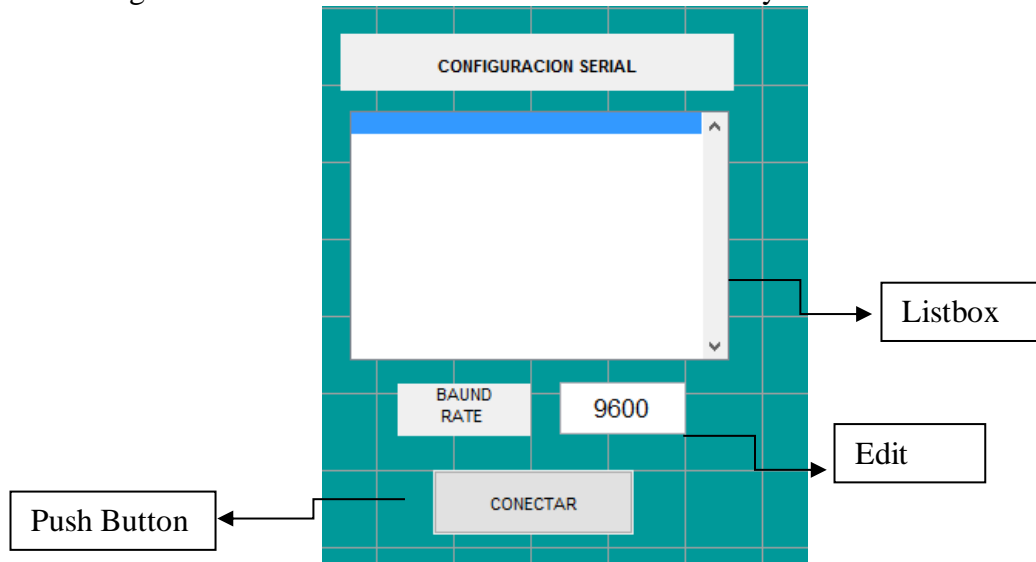
```

Código para establecer la comunicación entre el Matlab y el Arduino mediante puerto serial. (The MathWorks, 2016)

En la primera línea de código se obtiene información relacionada al hardware con el que se está conectado (Puerto COM). La segunda línea de código se obtiene la longitud del puerto serial y por último la tercera línea permite asignar mediante identificadores *handles* un puerto del Arduino, estos identificadores llaman a funciones indirectamente.

Para la configuración serial entre el Matlab y el Arduino se utilizó los siguientes componentes:

Figura 3.18 Interfaz de comunicación entre Matlab y el Arduino.



Mediante la configuración serial se puede comunicar entre el Matlab y el Arduino. (The MathWorks, 2016)

- a) **Listbox** en este componente se mostrará el puerto al cual está conectado el Arduino.
- b) **Edit Text** muestra la velocidad de comunicación que existe entre el Matlab y el Arduino.
- c) **Push Button** se encuentra la comunicacion entre el Arduino y el Matlab.

Se utiliza el comando *strcmp* para comparar dos string (cadena de caracteres), en este caso se compara el nombre colocado en el boton con la palabra *CONECTAR*. El identificador *handles* llama a la funcion *portlist* y obtiene un valor de 0 ó 1 del puerto serial al cual esta conectado el Arduino, este valor es guardado en una variable *serPortn*. Si el valor obtenido es 1 se envia un mensaje de error solicitando la

selección de un puerto válido; caso contrario el identificador obtiene el valor de la función *portlist* en una *matriz de celdas*, que se guardará en la variable *serList*. Es necesario convertir la *matriz de celdas* a una cadena de caracteres, que se guarda en la variable *serPort*.

El comando *serial* obtiene como primer atributo el puerto que se encuentra disponible, el segundo atributo da un tiempo de espera de 1 segundo. Para completar la lectura de dicho puerto, el último atributo obtiene la velocidad de comunicación de la función *baudrate*, convirtiéndolo de un *String* a número con el comando *str2num*.

Dentro de la sentencia *try* se obtiene información de la variable *serConn* mediante el comando *fopen*, que es asignado a un identificador para ser llamado en otras funciones.

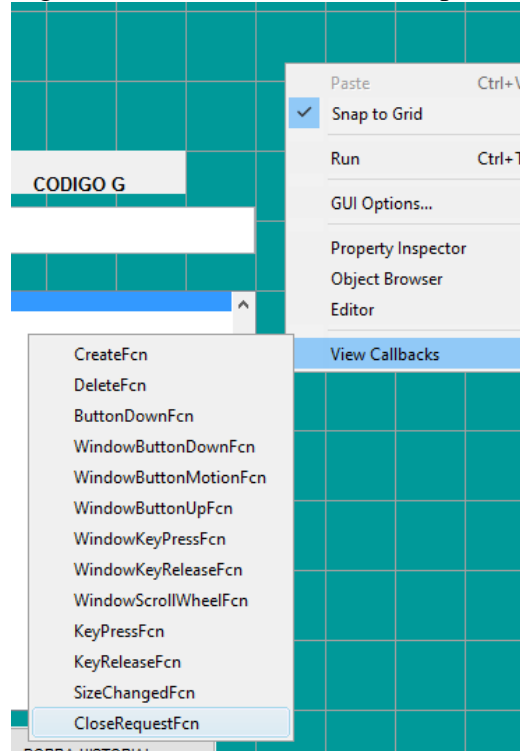
La línea de código `set(handles.Tx_send, 'Enable', 'On')` habilita la función *Tx_send* para el ingreso de comandos desde teclado por el usuario. Por otro lado, la línea de código `set(hObject, 'String','DESCONECTAR')` cambia el nombre asignado al botón.

En la sentencia *catch* captura un mensaje de error generado al abrir el puerto serial, imprimiendo un mensaje de solicitud de puerto. Si no se cumple la configuración inicial, se deshabilita la función *Tx_send* y se cambia el nombre del botón a CONECTAR. Finalmente se cierra el puerto de comunicación mediante el comando *fclose*.

d) Cierre de Puerto serial

Dentro de la interfaz se crea la función *CloseRequestFcn*, dando clic derecho sobre la interfaz como se muestra en la Figura 3.19.

Figura 3.19 Ventana de cierre de puerto.



CloseRequestFcn para el cierre de la interfaz. (The MathWorks, 2016)

Esta función permite deshabilitar el puerto serial en caso de existir un cierre inesperado de la interfaz. Esto se logra mediante la siguiente línea de código.

Figura 3.20 Código *CloseRequestFcn*

```

381 - function figure1_CloseRequestFcn(hObject, eventdata, handles)
382 -     if isfield(handles, 'serConn')
383 -         fclose(handles.serConn);
384 -     end
385 -     delete(hObject);

```

CloseRequestFcn cierre de un puerto en ejecución. (The MathWorks, 2016)

La sentencia *if* verifica mediante el comando *isfield* si existe alguna función con el nombre *serConn* ejecutándose, si cumple con la condición cierra el puerto mediante el comando *fclose*.

- **Paso 5.** Adquirir una imagen vectorizada.

Mediante la utilización de un Pushbutton se puede adquirir una imagen que se encuentra dentro de una carpeta ubicada en el mismo repositorio de los archivos de la GUI. Dentro de la función *Fig_1_Callback* se programa cada una de las líneas de código para dicha adquisición.

Figura 3.21 Código Pushbutton.

```
% --- Executes on button press in fig_1.
function fig_1_Callback(hObject, eventdata, handles)
Dat = fopen('J:\Desktop\fotos tesis\MATLAB1\INTERFAZ_FINAL\Codigo_G\calabaza2.ngc','r');
n=1;

while ~feof(Dat)
    leer_linea = fgetl(Dat);
    assignin('base','leer_linea',leer_linea)
    pause(2.01);
    C(n)= cellstr(leer_linea);
    n=n+1;
    if ~feof(Dat)
        fprintf(handles.serConn, leer_linea);
        set(handles.Historial, 'String',C);
        m=(n*100)/240;
        k=fix(m);
        p=num2str(k);
        porc=strcat(p,'%');
        set(handles.avance_1,'String',porc);
    end
end
fclose(Dat);
C=C';
```

Líneas de código para programación del pushbutton. (The MathWorks, 2016)

Se asigna a la variable *Dat* el comando *fopen* para abrir el archivo y especificar si es de lectura o escritura, en este caso solo es de lectura.

En la sentencia *while* se utiliza el comando *feof* para leer el archivo completo este especifica si el archivo a sido leído completamente enviando un valor de 1, si el archivo no se ha terminado de leer envia un valor de 0.

El comando *fgetl* lee cada línea del archivo excluyendo la siguiente y se la guarda en la variable *leer_linea*. Esta variable es asignada a un workspace convirtiendola en una variable local, se asigna un tiempo de espera para la lectura de la línea .

El comando *cellstr* obtiene del workspace la variable local y la convierte una matriz de caracteres guardandola en una variable *C(n)*, para ser mostrado en la funcion *historial*.

Se crea un contador *n* para almacenar cada línea del archivo en la matriz *C(n)*, esta variable se inicializa en 1.

Mediante el comando *fprintf* imprime cada línea del archivo en la funcion *serConn* de la comunicación serial. Al finalizar la sentencia *while*, se debe cerrar el archivo mediante el comando *fclose*.

Para que el archivo se imprima consecutivamente en el historial se utiliza el siguiente código `C=C';`.

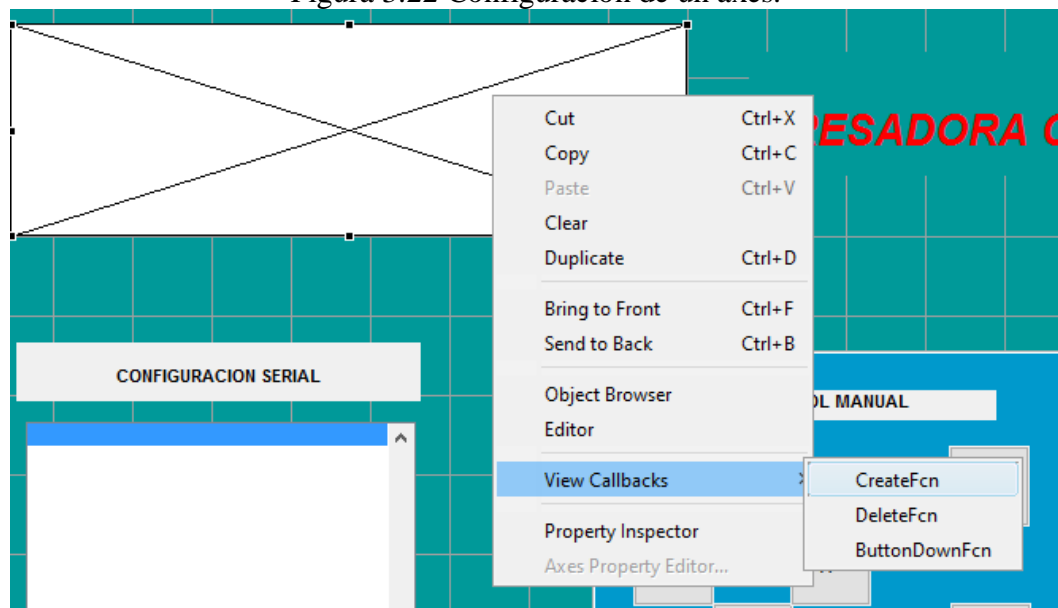
a) Borrar historial

Para borrar el contenido del historial se crea un botón con la siguiente configuración:
`set(handles.historial, 'string', '');`. Esta línea envía un espacio en blanco a la función historial, borrando su contenido.

b) Adquisición de imágenes mediante axes

En el archivo `.fig` se selecciona el componente axes, este crea un bloque, al dar un clic derecho sobre el se puede acceder a las propiedades. Se selecciona *View Callback* para crear una función *Fcn* en el archivo `.m`.

Figura 3.22 Configuración de un axes.



Archivo `m`. para configuracion de un axes. (The MathWorks, 2016)

Dentro de la función se configura la siguiente línea de código:

Figura 3.23 Código Axes

```
389 - function axes2_CreateFcn(hObject, eventdata, handles)
390 -     a=imread('calabaza2.jpg');
391 -     image(a)
392 -     axis off
```

Líneas de código para mostrar imagen en axes. (The MathWorks, 2016)

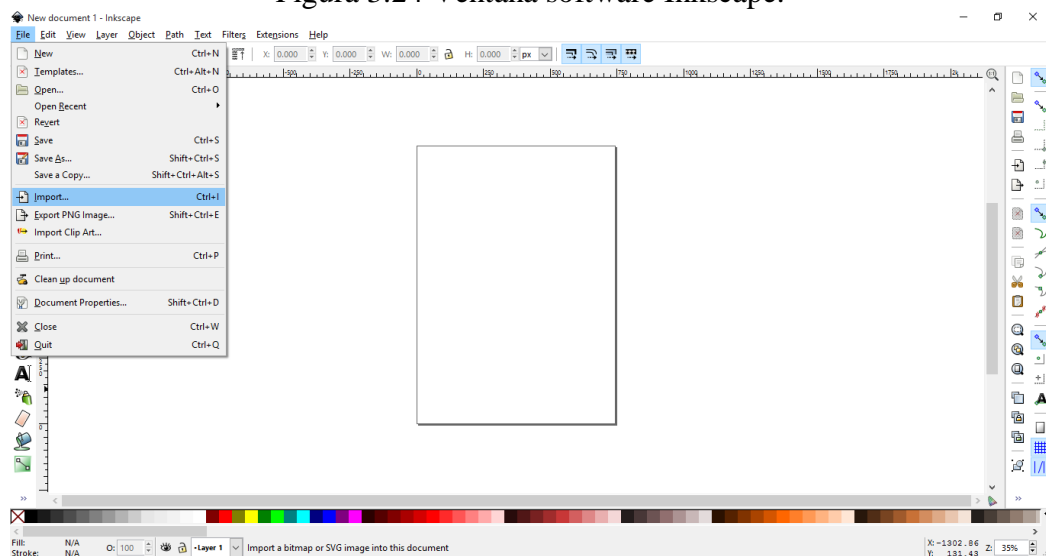
El comando *imread* lee el archivo *.jpg* y lo guarda en la variable *a*. El comando *image* muestra la imagen seleccionada en la interfaz. El comando *axis* permite habilitar y deshabilitar los ejes de coordenadas del componente axes.

3.5. Vectorización de Imagen *.jpg*.

Para la vectorización de imágenes en Código g (código con el cual trabaja la CNC), se utiliza un software libre llamado *Inkscape*. Este software ayuda a vectorizar imágenes, además utiliza el formato SVG (Scalable Vector Graphics). Dicho software es compatible con Windows, Linux y Mac OS X y otros más.

Primero se debe descargar el software desde la siguiente página (<https://inkscape.org/es/release/0.92.2/>) inmediatamente se abre el software, se despliega la siguiente pantalla.

Figura 3.24 Ventana software Inkscape.

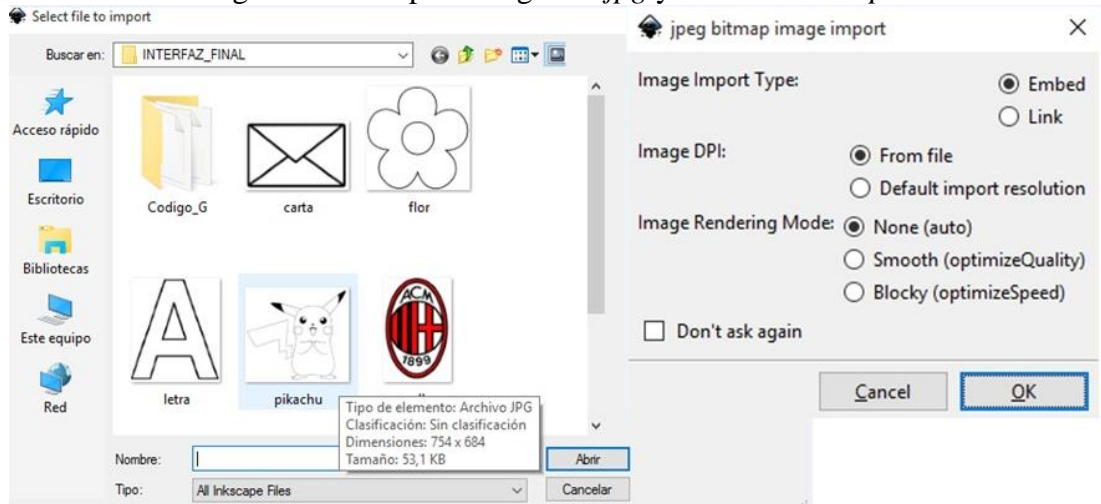


Ventana Inkscape permite la vectorización de imágenes *.jpg*. (Inkscape, 1999)

A continuación, se debe de ir a la pestaña “File” y en la opción *import* se selecciona el archivo *.jpg* para importar la imagen que se va a vectorizar y crear el Código G.

Cuando se haya seleccionado la imagen aparecerá una pantalla, en esta no se debe ejecutar ningún cambio solo se presiona *OK* y mostrará la imagen en la interfaz principal de *Inkscape*.

Figura 3.25 Carpeta imágenes .jpg y ventana *Inkscape*.



Dentro de la carpeta de programación se guarda cada uno de las imágenes a vectorizar para ser importadas. (Inkscape, 1999)

Figura 3.26 Imagen seleccionada.

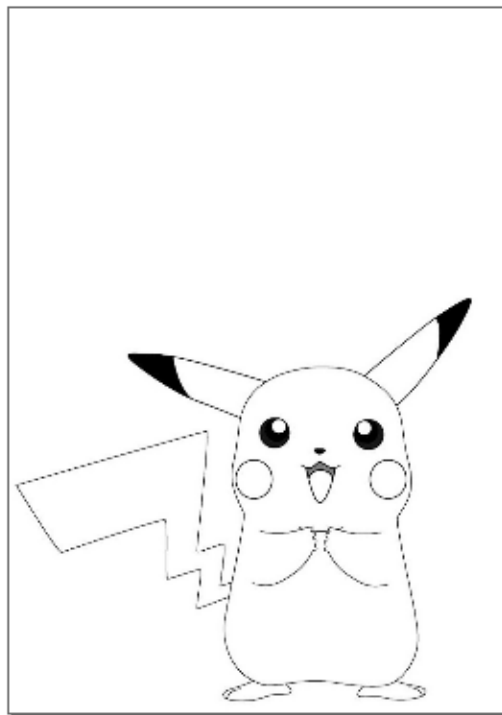
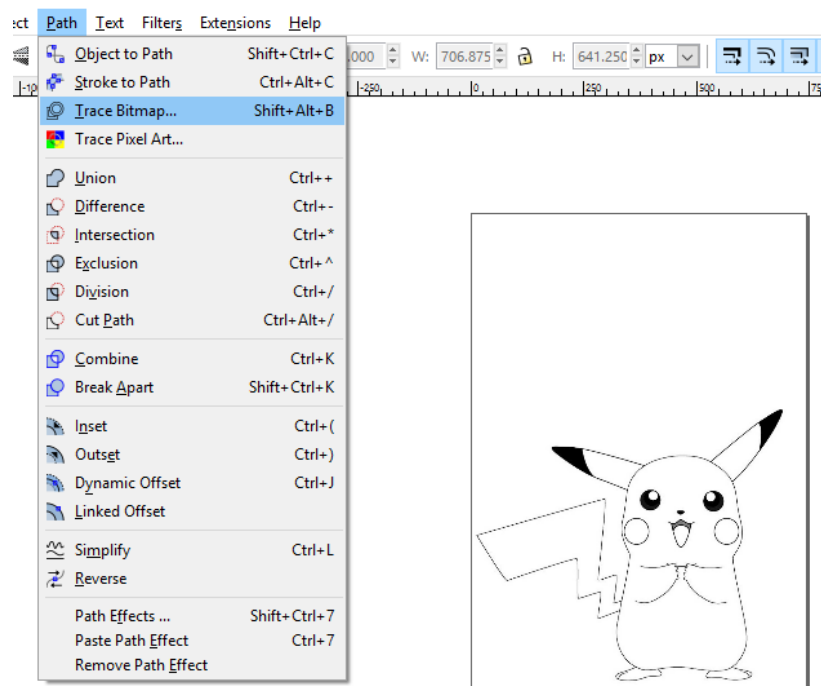


Imagen seleccionada de la carpeta para ser vectorizada. (Inkscape, 1999)

Luego se debe de seleccionar la imagen e inmediatamente se presiona la pestaña *Path* y se elige la opción *Trace Bitmap*.

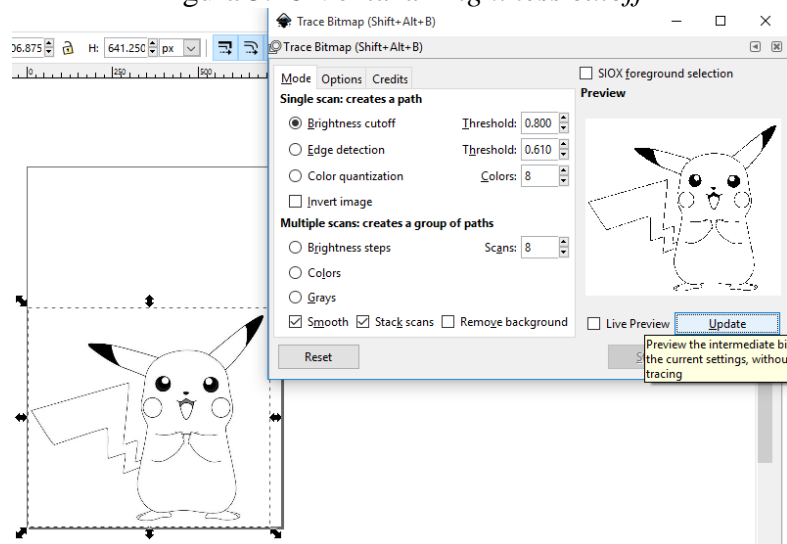
Figura 3.27 Ventana *Trace Bitmap*.



Trace Bitmap permite configurar la imagen para ser vectorizada. (Inkscape, 1999)

Se puede ajustar el contraste de la imagen, en la opción *Brightness cutoff*, para que sea más clara y se pueda observar los bordes de la misma. Cuando se obtenga la imagen deseada se da clic en *Update*, y se creará una nueva imagen vectorizada.

Figura 3.28 Ventana *Brightness cutoff*



Brightness cutoff es una opción de la venta *trace bitmap* que permite optimizar la imagen antes de vectorizar. (Inkscape, 1999)

A continuación, se creará una imagen de puntos color azul (esta será la imagen que se ha vectorizado), mientras que la otra imagen se debe de borrar para que no se confunda en el proceso.

Figura 3.29 Imagen vectorizada.

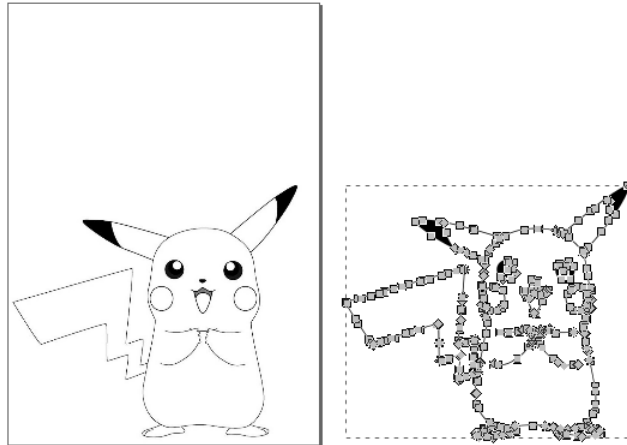
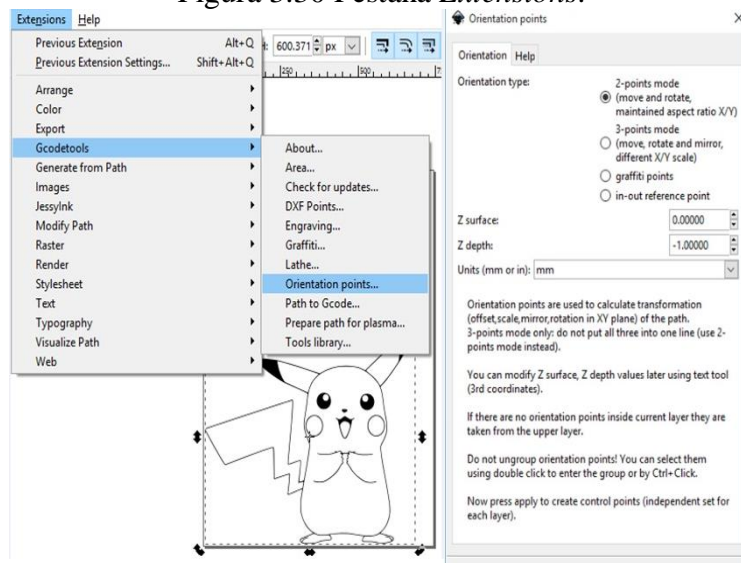


Imagen derecha vectorizada, imagen izquierda seleccionada. (Inkscape, 1999)

Como se mencionó anteriormente se va a trabajar sobre la imagen vectorizada, en la pestaña *path* se debe de seleccionar las opciones *Object to Path* y *Dynamic Offset*.

Después de haber realizado el paso anterior se debe dirigir hacia la pestaña *Extensions*, en la opción *Gcodetools* y luego *Orientation Points*. Para esto, se debe de seleccionar la imagen antes de realizar algún cambio o algún procedimiento. Luego aparecerá una ventana adicional para configurar el recorrido del eje *z* para el tallado de las piezas. En la opción (*Z surface*) se configura la profundidad inicial del corte, en la siguiente opción (*Z depth*) se configura la profundidad total de corte, la profundidad está dada en milímetros [mm] y finalmente se dará clic en *Apply*.

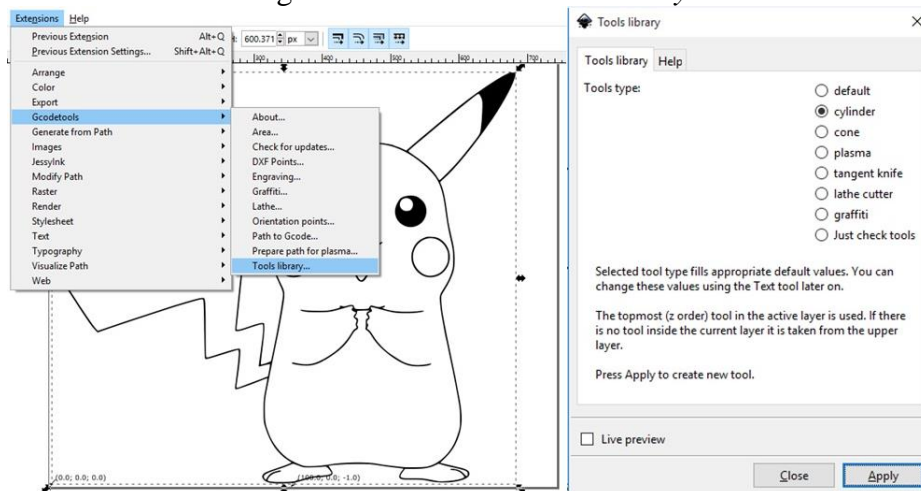
Figura 3.30 Pestaña *Extensions*.



Ventana *Extensions* permite configurar cada uno de los puntos de orientación de la imagen. (Inkscape, 1999)

Al momento de dar clic en *Apply* en el plano aparecerá unas coordenadas que representan los puntos de orientación de la imagen. Nuevamente en la pestaña *Extensions*, se elige la opción *GcodeTools* y luego *Toolslibrary*, entonces aparecerá una ventana, esta servirá para seleccionar el tipo de fresa que se va a utilizar para el tallado de la pieza.

Figura 3.31 Ventana *Tools library*



Tools library permite configurar la fresa que se va a utilizar. (Inkscape, 1999)


Después de dar clic en *Apply*, aparecerá un cuadro de texto donde se especifica el tipo de fresa, el diámetro de la misma, la velocidad a la cual va a trabajar, el ángulo y la velocidad a la cual se va a perforar en el material de tallado Figura 3.32. Para modificar esta variables se debe de seleccionar la opción de *edit text* .

Figura 3.32 Especificaciones técnica del fresado.

name	Cylindrical cutter
id	Cylindrical cutter 0001
diameter	3.2
feed	400
penetration angle	90
penetration feed	100
depth step	1
tool change gcode	(None)

Especificaciones del diámetro y ángulo de penetración software *Inkscape*. (Inkscape, 1999)

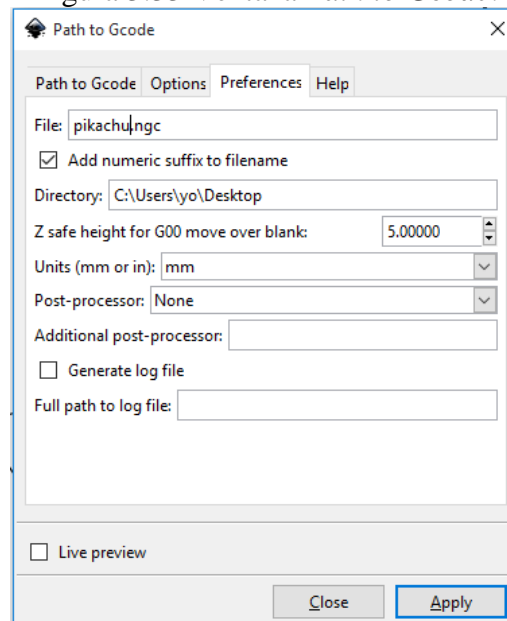
- *Diameter*: Se configura el diámetro de la fresa con la cual se trabaja.
- *Feed*: Se configura la velocidad de tallado.

- *Penetration angle*: Se configura el ángulo en que la fresa se introduce en el material, este valor se deja el valor por defecto
- *Penetration feed*: Se configura la velocidad a la cual la fresa se introduce en el material.
- *Depth step*: Se configura los pasos de avance de la profundidad de la fresa.

Para obtener el Código G de la imagen, en la pestaña *Extensions*, se selecciona la opción *Gcodetools* y luego *Path to Gcode*.

En la ventana *Path to Gcode*, se ingresa a la pestaña *Preferences*, en esta parte se debe colocar el nombre con el cual se va a guardar el archivo *.ngc* y además se debe especificar el directorio donde se guardará la imagen. También se puede modificar el punto de inicio del eje Z, en este caso se da un valor de 5.

Figura 3.33 Ventana *Path to Gcode*.

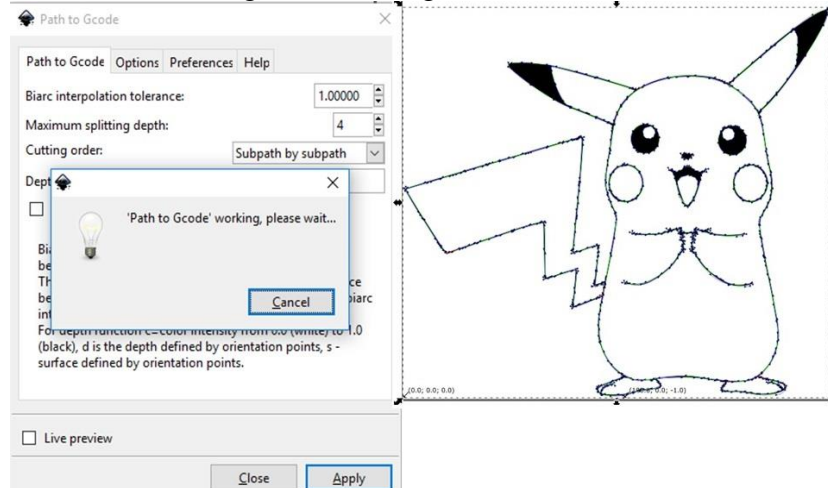


Path to Gcode se especifica el directorio en el cual se guarda la imagen *.ngc* (Inkscape, 1999)

Cuando se haya terminado de realizar las acciones anteriores se ingresa en la pestaña *Path to Gcode*. En esta parte no se debe de realizar algún cambio, se da clic en *Apply*. Cabe mencionar que este paso es importante ya que si se da clic cuando esté en la pestaña *Preferences* dará error. Por eso es recomendable estar en la pestaña *Path to Gcode*.

Este proceso tardara unos minutos, cuando finalice de cargar en los bordes la imagen se puede observar unas flechas que indican el camino por donde va a trabajar la fresa.

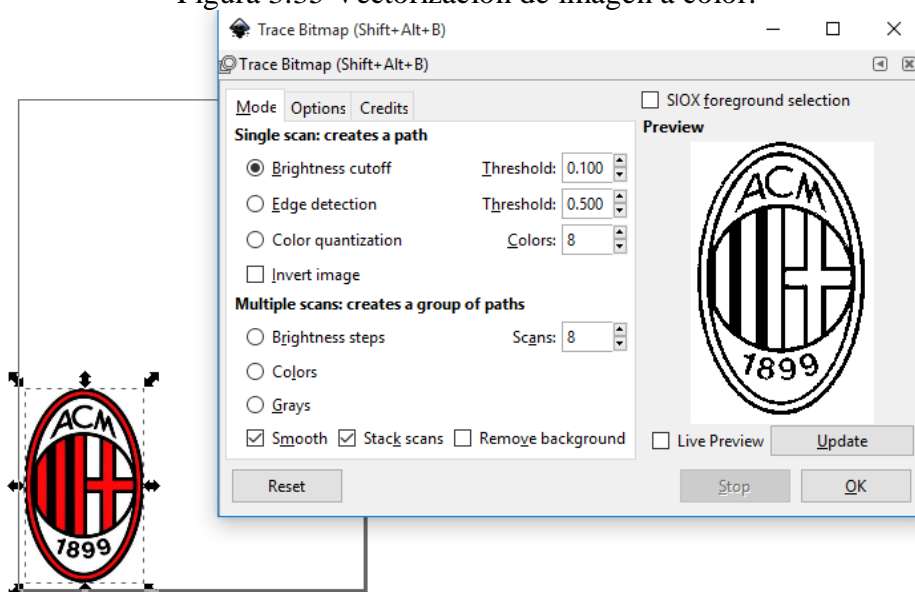
Figura 3.34 Imagen vectorizada.



La imagen a la derecha esta vectorizada indicando con flechas la dirección de trazado. (Inkscape, 1999)

Es necesario mencionar que este proceso es aplicado a cualquier tipo de imagen. Es necesario modificar la opción *Brightness cutoff*, para mejorar la resolución de la imagen. Mientras el valor sea menor la imagen será más clara y se podrá observar los bordes con una mejor resolución. Cuando se obtenga la imagen deseada se da clic en *Update* y *OK*, entonces se creará la nueva imagen vectorizada.

Figura 3.35 Vectorización de imagen a color.



Se puede observar que todas las imágenes *jpg*, tanto en blanco y negro como a color se vectoriza sin problemas.(Inkscape, 1999)

No se debe de olvidar que durante el proceso siempre se debe seleccionar la imagen con la que vamos a trabajar (que es la imagen vectorizada).

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Este capítulo permite conocer mediante pruebas el correcto funcionamiento del sistema de automatización de la fresadora CNC, se tomará como principal factor la comparación de tiempos, entre la máquina manual y el sistema de automatización propuesto. Mediante una encuesta se evaluó el grado de afectación del polvo, el desperdicio de la materia prima, el ruido ejercido por cada una de las máquinas y la seguridad física del operario.

4.1. Prueba de tiempo de tallado de pieza en madera

El procedimiento para poner en marcha el sistema de automatización de la fresadora CNC se basa en los siguientes pasos:

- a. Conectar la interfaz de Matlab al Arduino.
- b. Determinar el punto cero de la máquina.
- c. Seleccionar la imagen a ser tallado.
- d. Iniciar el tallado.

A partir de estos pasos se tomó los siguientes tiempos de tallado de la pieza de madera que se muestra en la Figura 4.1.

Figura 4.1 Tallado de madera con la fresadora CNC.



Imagen tallada con sistema de automatización de la fresadora CNC. Elaborado por: Pilatuña & Pillajo.

Tabla 4.1 Tiempos de tallado con sistema de automatización.

Número de Pruebas	Tiempos de tallado (h)
Prueba 1	0,038
Prueba 2	0,042
Prueba 3	0,043
Prueba 4	0,033
Prueba 5	0,038
Prueba 6	0,038
Prueba 7	0,038
Prueba 8	0,038
Prueba 9	0,038
Prueba 10	0,038

Tiempos en horas para comparación con máquina manual. Elaborado por: Pilatuña & Pillajo

Para realizar el tallado con el sistema anterior de control de la empresa, el operario necesita de una plantilla para ejecutar dicho trabajo. Esta plantilla tiene un tiempo de confección alrededor de 12 horas, tiempo que se demora para obtener una plantilla con los trazos requeridos por el usuario.

A partir de este proceso se tomó los tiempos de tallado manual como se muestra en la Tabla 4.2.

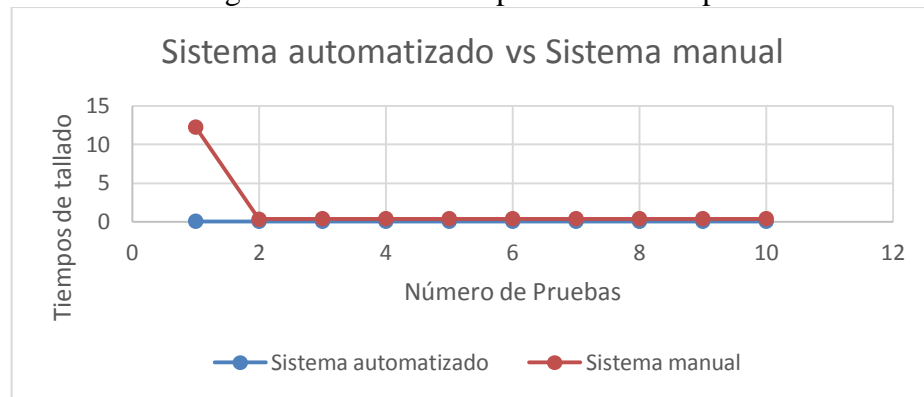
Tabla 4.2 Tiempo de tallado realizado por el operario

Número de Pruebas	Tiempos de tallado (h)
Prueba 1	12,30
Prueba 2	0,35
Prueba 3	0,36
Prueba 4	0,38
Prueba 5	0,38
Prueba 6	0,39
Prueba 7	0,38
Prueba 8	0,38
Prueba 9	0,38
Prueba 10	0,39

Tiempos de pruebas realizados manualmente por el operario. Elaborado por: Pilatuña & Pillajo

El tiempo referente a la prueba 1 corresponde a la confección de la plantilla y el tiempo de tallado realizado por el operario.

Figura 4.2 Gráfica comparativa de tiempos



Esquema comparativo de tiempos de tallado de madera del sistema automático y manual. Elaborado por: Pilatuña & Pillajo

La Figura 4.2 muestra la gráfica comparativa entre los tiempos de tallado tanto manual como automatizado, indicando un ahorro en tiempo en el proceso de tallado con el sistema automatizado, ya que este sistema no necesita de plantilla para ejecutar el trabajo. El tallado manual necesita de más tiempo para obtener un buen trabajo final.

4.2. Prueba de tiempo de cortado de pieza en madera

El procedimiento para poner en marcha el sistema de automatización de la fresadora CNC se basa en los siguientes pasos:

- Conectar la interfaz de Matlab al Arduino.
- Determinar el punto cero de la máquina.
- Seleccionar la imagen a ser cortado.
- Iniciar el corte.

A partir de estos pasos se tomó los siguientes tiempos de tallado de la pieza de madera que se muestra en la Figura 4.3.

Figura 4.3 Corte de madera con la fresadora CNC.



Imagen cortada con sistema de automatización de la fresadora CNC. Elaborado por: Pilatuña & Pillajo

Tabla 4.3 Tiempos de pruebas de corte en piezas de madera

Número de Pruebas	Tiempos de corte (h)
Prueba 1	0,069
Prueba 2	0,071
Prueba 3	0,072
Prueba 4	0,071
Prueba 5	0,072
Prueba 6	0,072
Prueba 7	0,072
Prueba 8	0,072
Prueba 9	0,072
Prueba 10	0,072

Tiempos en horas para comparación con máquina manual. Elaborado por: Pilatuña & Pillajo

Para realizar el corte con el sistema anterior de control de la empresa, el operario necesita de una plantilla para ejecutar dicho trabajo. Esta plantilla tiene un tiempo de confección alrededor de 12 horas, tiempo que se demora para obtener una plantilla con los trazos requeridos por el usuario.

A partir de este proceso se tomó los tiempos de corte manual como se muestra en la Tabla 4.4.

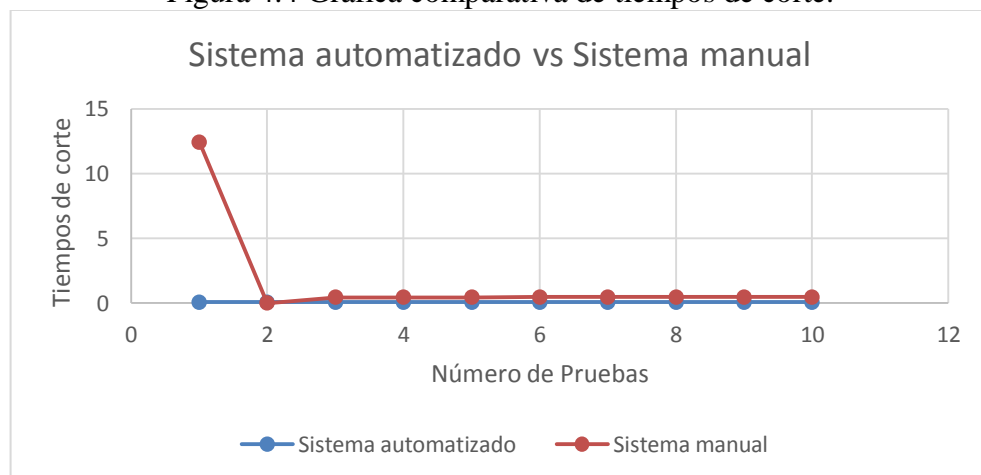
Tabla 4.4 Tiempo de corte realizado por el operario.

Número de Pruebas	Tiempos de tallado (h)
Prueba 1	12,438
Prueba 2	0,435
Prueba 3	0,450
Prueba 4	0,458
Prueba 5	0,458
Prueba 6	0,460
Prueba 7	0,466
Prueba 8	0,466
Prueba 9	0,467
Prueba 10	0,467

Tiempos de pruebas realizados manualmente por el operario. Elaborado por: Pilatuña & Pillajo

El tiempo referente a la prueba 1 corresponde a la confección de la plantilla y el tiempo de corte realizado por el operario.

Figura 4.4 Gráfica comparativa de tiempos de corte.



Esquema comparativo de tiempos de corte de madera del sistema automático y manual. Elaborado por: Pilatuña & Pillajo

La Figura 4.4 muestra la gráfica comparativa entre los tiempos de corte tanto manual como automatizado, indicando un ahorro en tiempo en el proceso de corte con el sistema automatizado, ya que este sistema no necesita de plantilla para ejecutar el trabajo. El corte manual necesita de más tiempo para obtener un buen trabajo final.

4.3. Discusión de resultados

A partir de los resultados obtenidos con las pruebas del sistema automático y el operario se realizó la encuesta que se muestra en la Tabla 4.5. El número de

encuestas es proporcional al número de pruebas realizadas, es decir, se realizó una encuesta por cada prueba de funcionamiento.

Tabla 4.5 Preguntas del funcionamiento de la Fresadora CNC.

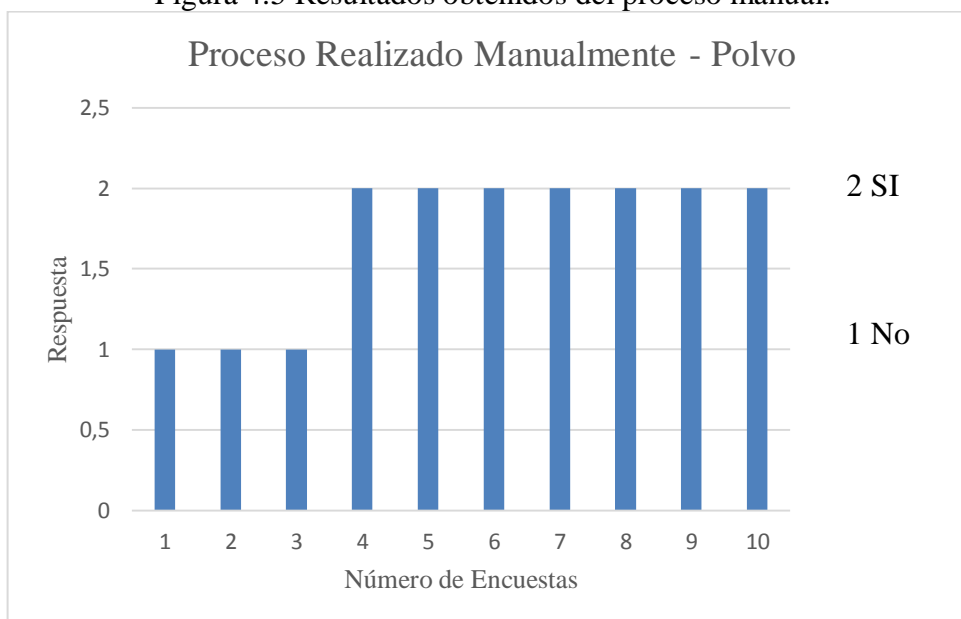
ITEM	ENCUESTA DE FUNCIONAMIENTO DE LA FRESADORA CNC.
1	¿Considera usted que al momento de realizar el tallado o corte de madera fue afectado por el polvo?
2	¿Considera usted que al momento de realizar el tallado o corte de madera se reduce el desperdicio de la materia prima?
3	¿Considera usted al momento de realizar el tallado o corte de madera fue afectado por el ruido de la maquina?
4	¿Considera usted que al momento de realizar el tallado o corte de madera fue afectado su seguridad física?

Elaborado por: Pilatuña & Pillajo

4.3.1. Resultados obtenidos de la pregunta 1

¿Considera usted que al momento de realizar el tallado o corte de madera fue afectado por el polvo?

Figura 4.5 Resultados obtenidos del proceso manual.



Gráfica de barras que representa resultados obtenidos del proceso manual, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

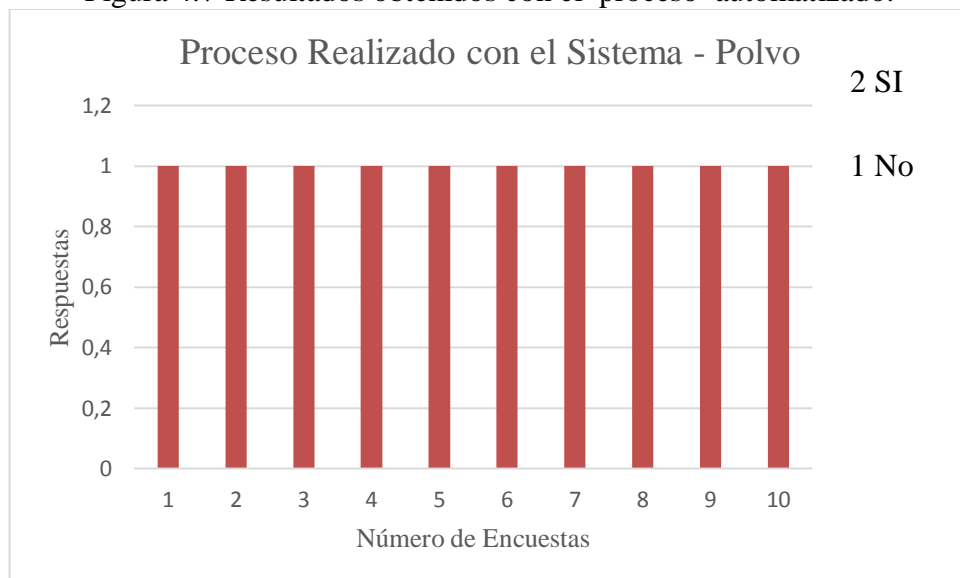
Figura 4.6 Restos de polvo obtenidos de tallado.



Residuos de polvo obtenido de tallado de pieza de madera. Elaborado por: Pilatuña & Pillajo

La empresa cuenta con una máquina fresadora manual que, al momento de realizar el corte o tallado de la pieza de madera esparce una gran cantidad de aserrín y polvo, el cual es perjudicial para la salud del operador en la parte respiratoria.

Figura 4.7 Resultados obtenidos con el proceso automatizado.



Gráfica de barras que representa resultados obtenidos del sistema automatizado, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

Figura 4.8 Aspiradora para eliminar restos de polvo y aserrín.



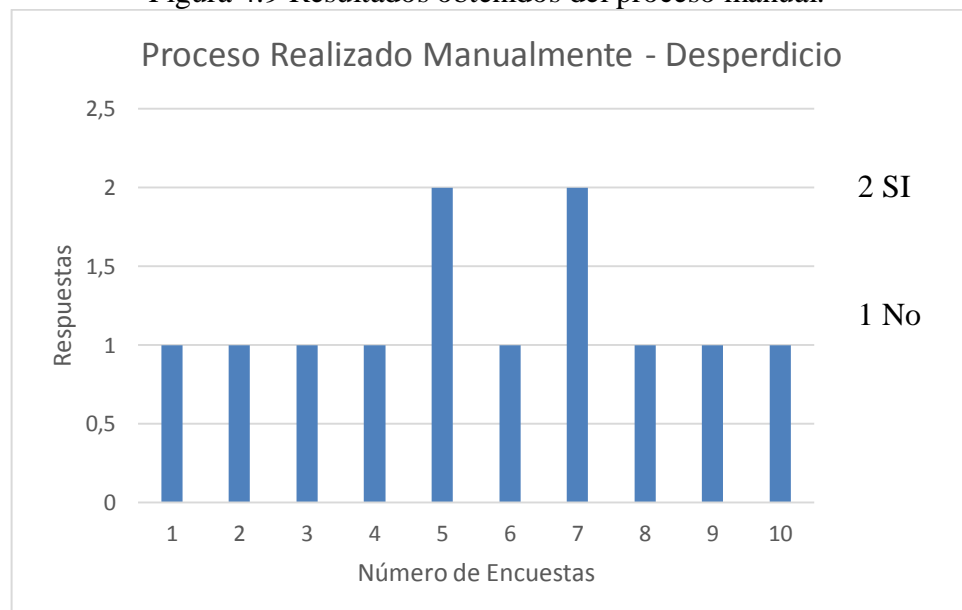
Equipo para la absorción de polvo y aserrín implementado en fresadora CNC. Elaborado por: Pilatuña & Pillajo

En el sistema actual de la maquina CNC se implementó una aspiradora que reduce el índice del polvo casi en su totalidad, permitiendo que el operador no sufra algún daño respiratorio y pueda conservar su salud.

4.3.2. Resultados obtenidos de la pregunta 2

¿Considera usted que al momento de realizar el tallado o corte de madera se reduce el desperdicio de la materia prima?

Figura 4.9 Resultados obtenidos del proceso manual.



Gráfica de barras que representa resultados obtenidos del proceso manual, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

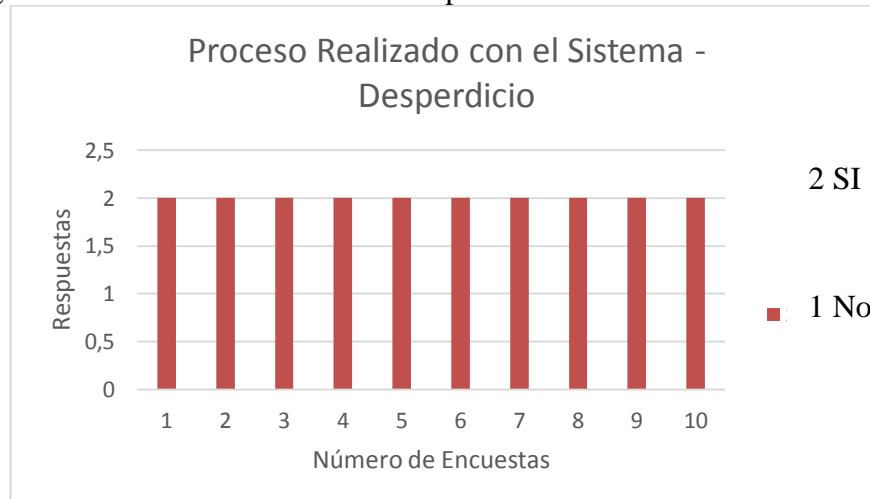
Figura 4.10 Tallado de pieza de madera realizada manualmente.



Tallado realizado por el operario mediante plantilla. Elaborado por: Pilatuña & Pillajo

Como se observa en la Figura 4.10 al momento de realizar el tallado de una pieza de madera existe algunas fallas con respecto a la imagen que se quiere obtener, ya que el operador manipula la pieza de madera y puede generar errores que baja la calidad del producto final, haciendo que la pieza sea desechada.

Figura 4.11 Resultados obtenidos del proceso con el sistema automatizado.



Gráfica de barras que representa resultados obtenidos del sistema automatizado, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

Figura 4.12 Tallado de pieza de madera realizada con el sistema automatizado.



Tallado realizado con el sistema de automatización. Elaborado por: Pilatuña & Pillajo

Como se observa en la Figura 4.12 el tallado de la pieza de madera no tiene ningún error ya que el operario no interviene en la manipulación de la misma, lo controla por medio de la computadora. Además, se puede realizar tallados de alta dificultad, como se observa en la Figura 4.13.

Figura 4.13 Tallado de pieza de madera de alta dificultad

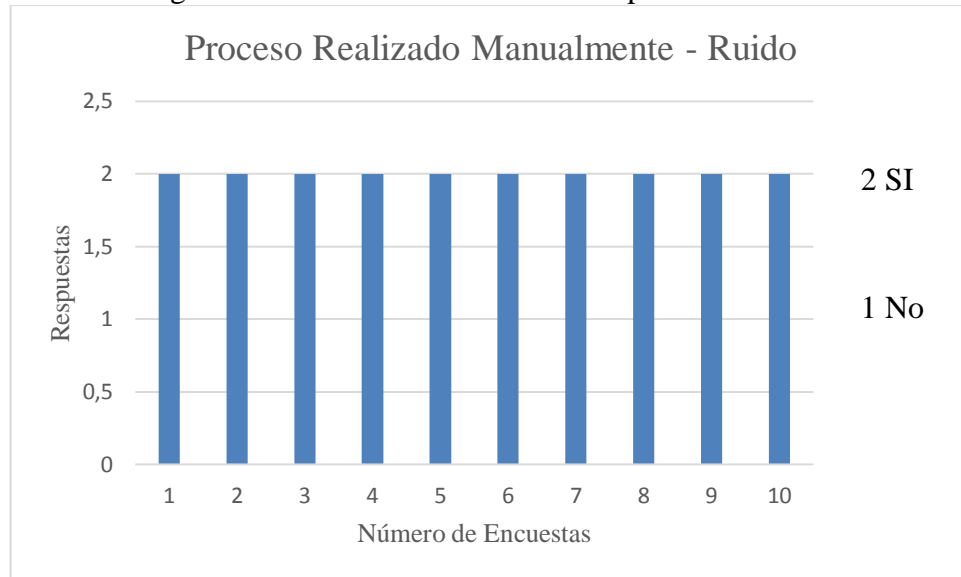


Tallado de pieza de madera de alta dificultad. Elaborado por: Pilatuña & Pillajo

4.3.3. Resultados obtenidos de la pregunta 3

¿Considera usted al momento de realizar el tallado o corte de madera fue afectado por el ruido de la máquina?

Figura 4.14 Resultados obtenidos del proceso manual.



Gráfica de barras que representa resultados obtenidos del proceso manual, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

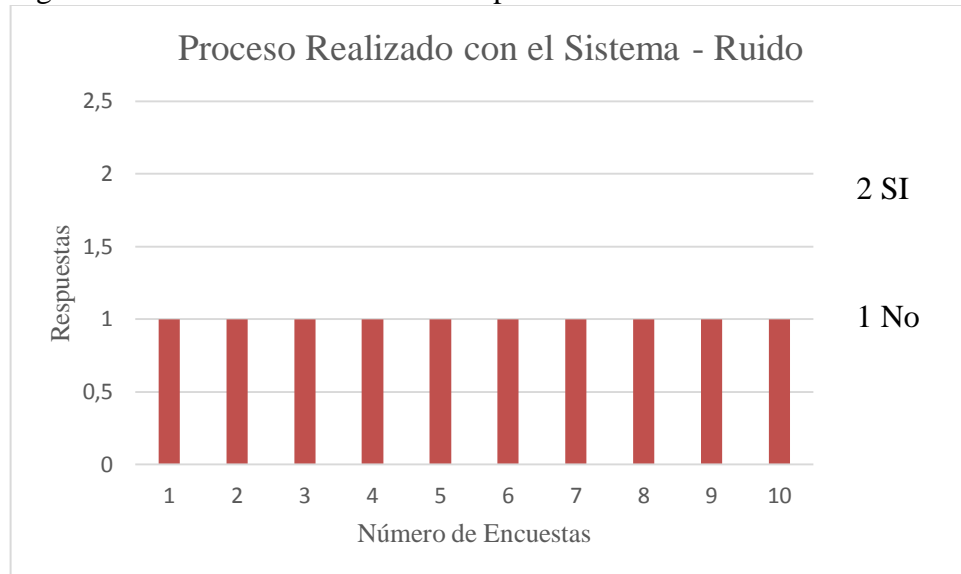
Figura 4.15 Máquina manual con un motor grande.



Equipo manual de tallado y corte de madera. Elaborado por: Pilatuña & Pillajo

La empresa cuenta con una máquina fresadora manual la cual tiene un motor grande que produce un ruido que afecta el sistema auditivo del operador, por eso se usa audífonos de protección, ya que a largo plazo puede existir pérdida auditiva.

Figura 4.16 Resultados obtenidos del proceso con el sistema automatizado.



Gráfica de barras que representa resultados obtenidos del sistema automatizado, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

Figura 4.17 Fresadora CNC con un motor pequeño.



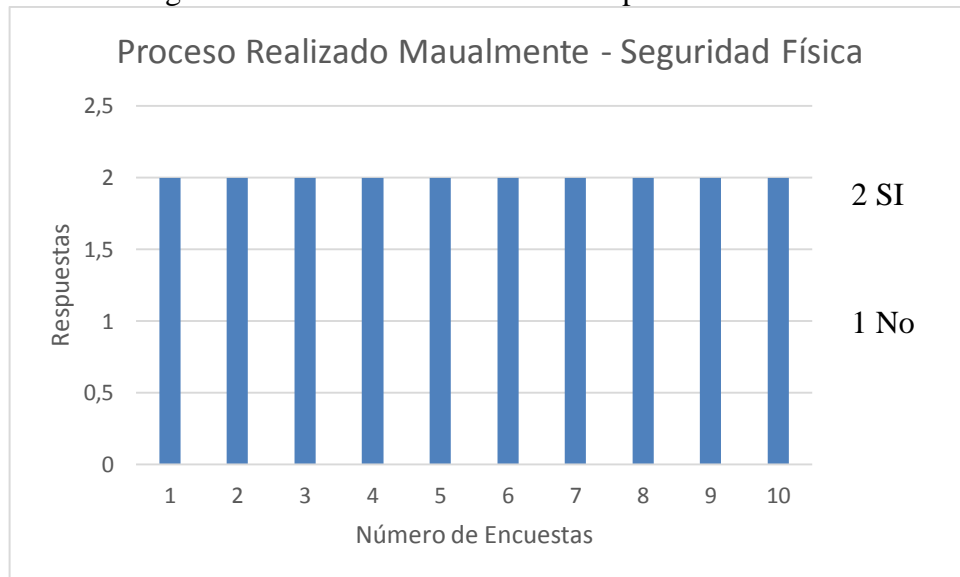
Fresadora CNC con motores paso a paso. Elaborado por: Pilatuña & Pillajo

En el sistema actual de la fresadora CNC el motor que se utiliza es más pequeño, entonces el ruido es mínimo, el operador no debe usar audífonos y no sufre de pérdida auditiva.

4.3.4. Resultados obtenidos de la pregunta 4

¿Considera usted que al momento de realizar el tallado o corte de madera fue afectado su seguridad física?

Figura 4.18 Resultados obtenidos del proceso manual.



Gráfica de barras que representa resultados obtenidos del proceso manual, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

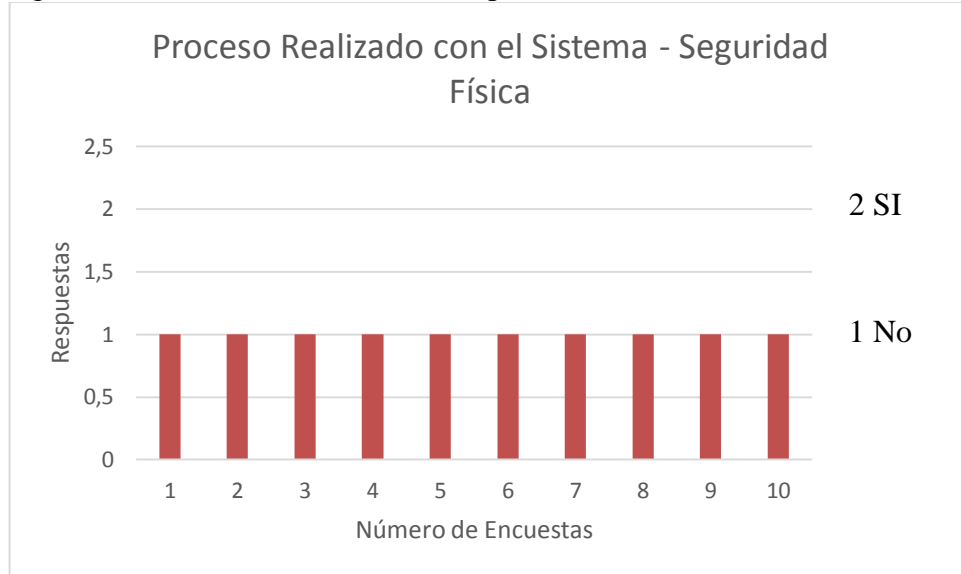
Figura 4.19 Operario realizando el corte manualmente.



Corte de pieza de madera realizada por el operario. Elaborado por: Pilatuña & Pillajo

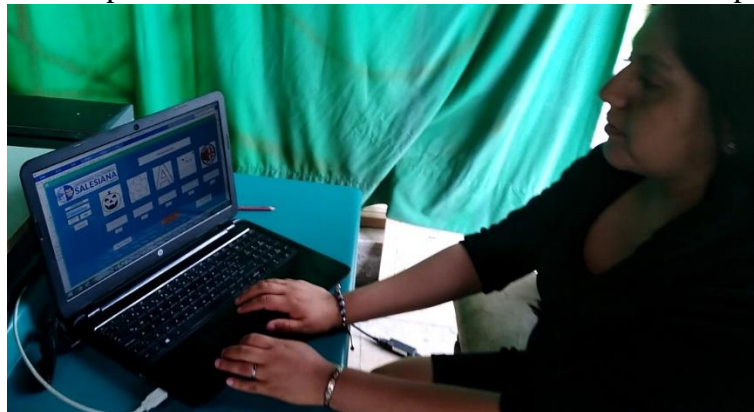
Al realizar el corte o tallado con la máquina fresadora manual el operario está expuesto a sufrir un accidente que podría comprometer su integridad física. Esto debido a que se debe manipular la pieza de madera con las manos en todo el proceso.

Figura 4.20 Resultados obtenidos del proceso con el sistema automatizado.



Gráfica de barras que representa resultados obtenidos del sistema automatizado, evaluados en 2 Si y en 1 No. Elaborado por: Pilatuña & Pillajo

Figura 4.21 Operario realizando el corte o tallado desde el computador.



Operario realizando tallado o corte de piezas de madera. Elaborado por: Pilatuña & Pillajo

Con el sistema actual de la fresadora CNC, el operario no interviene físicamente en la manipulación de la pieza de madera, entonces se puede proteger su integridad física y obtener un mejor resultado en el producto final.

CAPÍTULO 5

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Se determinó que se puede construir una máquina fresadora CNC utilizando una GUI de Matlab para controlar motores paso a paso mediante una tarjeta Arduino y a través de conexión serial. Se verificó su correcto funcionamiento al realizar pruebas repetitivas en corte y tallado de piezas de madera.

Con las pruebas realizadas se verificó la reducción de tiempo en proceso de producción alrededor de 5 minutos. Además, se comprobó también la reducción de desperdicio de material, disminución de ruido, polvo y los riesgos de seguridad física en el operario.

Se consiguió un software de control mediante una GUI de Matlab que permite que el usuario manipule el proceso de forma remota mediante comunicación serial. Dicha interfaz gráfica permite seleccionar una imagen para tallar o corta en una pieza de madera, todo esto gracias a un menú que se puede utilizar mediante un simple clic.

Las ecuaciones implementadas permitieron la adquisición de los motores paso a paso y los drivers de control. Estas determinaron el torque y el número de pasos con las que se trabajó.

Debido a las pruebas de funcionamiento que se realizaron se determinó que el sistema no necesita de un control retroalimentado, ya que el tallado y corte de madera no requiere de precisión milimétrica.

La máquina fresadora CNC puede comprender y ejecutar las líneas de código G generados por el software Inkscape, y esto se pudo comprobar con las pruebas realizadas al enviar el código al Arduino el cual permitió obtener el producto deseado sin error de tallado o corte.

Con el sistema de automatización implementado en la fresadora CNC se descartó el diseño del molde utilizado en el tallado o corte de piezas de madera, lo que permitió reducir el tiempo de trabajo invertido en el mismo.

5.2. Recomendaciones

Es recomendable verificar que cada driver tenga la misma configuración en número de pasos, para que el tallado o corte de pieza este de acuerdo a las medidas de la imagen vectorizada.

Para la adquisición de cada motor es recomendable realizar el cálculo respectivo a la velocidad y potencia con la que cada motor debe trabajar.

Hay que tener en cuenta las especificaciones técnicas de los elementos eléctricos y electrónicos, ya que puede generar daños a los drivers de control y al Arduino haciendo que su desempeño sea defectuoso.

Para realizar tallados o cortes de diferentes diseños, es necesario adquirir las fresas adecuadas para cada tipo de diseño, por ejemplo, para el tallado de letras se necesita fresas de diámetro pequeño.

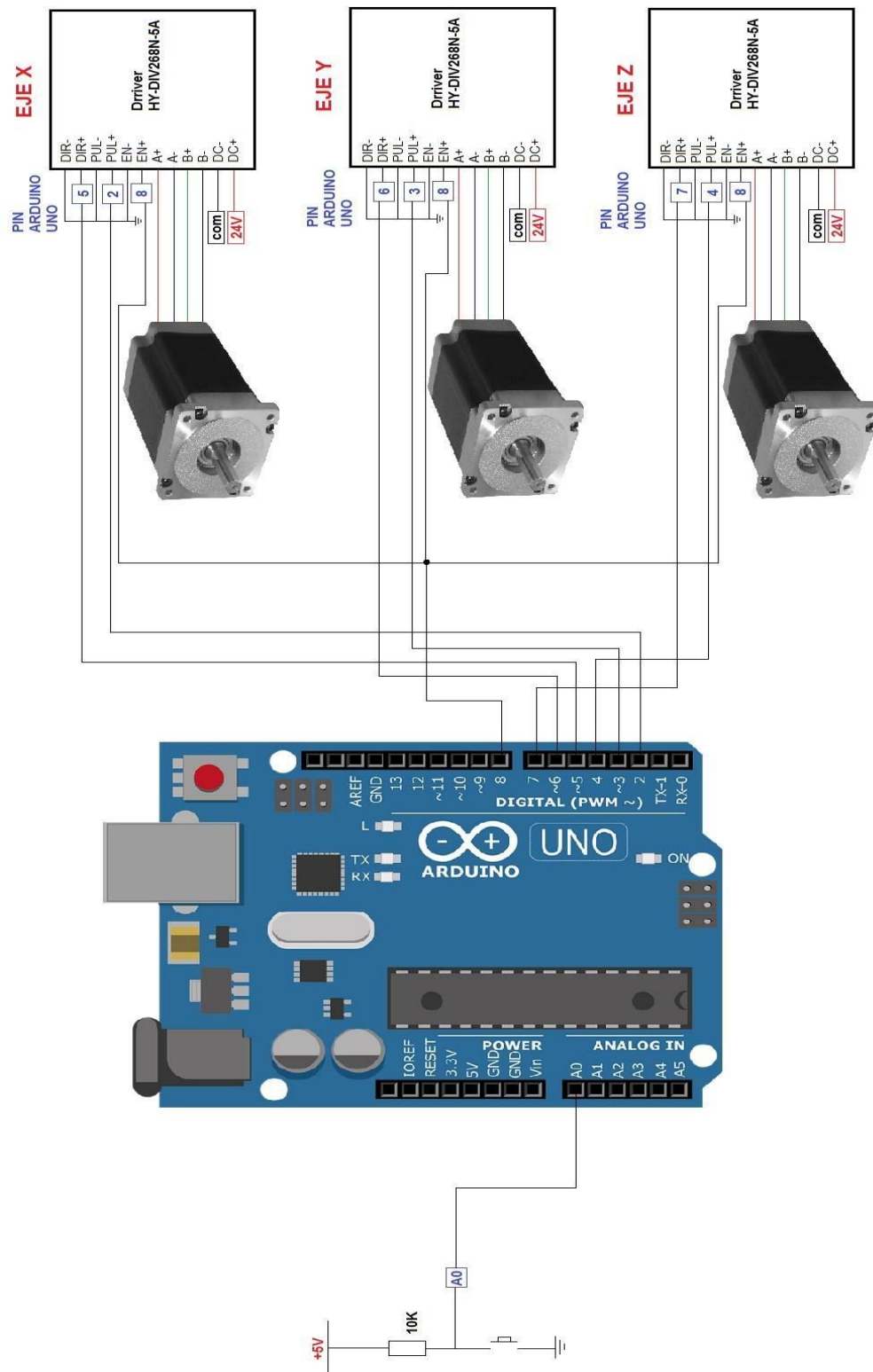
REFERENCIAS

- Arduino. (2018). *Software*. Obtenido de Arduino 1.8.5:
<https://www.arduino.cc/en/Main/Software>
- Bricogeek, T. (2013). Motor Paso a Paso Nema 25. Europa.
- Carletti, E. J. (2007). *Robots*. Obtenido de http://robots-argentina.com.ar/MotorPP_basico.htm
- Changzhou Wantai. (2009). Motor de pasos 57BYGH115-003. Changozhou, China.
- Do, M. (05 de Septiembre de 2014). *Marcenaria Do Edinho*. Obtenido de <http://www.marcenariadoedinho.com.br/mdf-ou-mdp-saiba-a-diferenca/>
- Electroniclab. (s.f.). *ELECRONICLAB*. Obtenido de Ingeniería y Diseños Electronicos: <https://electronilab.co/tienda/driver-microstepper-motor-paso-paso-tb6600-0-2-5a-12-48vdc/>
- (24 de Marzo de 2018). Encuesta de funcionamiento de la fresadora CNC. (Y. Pilatuña, & I. Pillajo, Entrevistadores)
- Gobierno de Canarias. (Mayo de 2013). *Ecoblog*. Obtenido de Características Arduino:
<http://www3.gobiernodecanarias.org/medusa/ecoblog/ralvgon/files/2013/05/Caracter%C3%ADsticas-Arduino.pdf>
- Hernandez, P. (10 de 12 de 2012). *ARQHYS*. Obtenido de <http://www.arqhys.com/casas/fresadoras-maquinas.html>
- Inkscape. (2 de Septiembre de 1999). Inkscape.
- Pilatuña, Y., & Pillajo, I. (05 de Marzo de 2018). Arquitectura para la automatización de la fresadora. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (05 de 2018). Corte de piezas de madera. Quito, Pichincha.
- Pilatuña, Y., & Pillajo, I. (05 de 2018). Discusión de encuestas. Quito, Pichincha.

- Pilatuña, Y., & Pillajo, I. (05 de Marzo de 2018). Indumentaria de Trabajo. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (30 de 04 de 2018). Interfaz GUI Matlab. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (05 de Marzo de 2018). Plantilla y Corte manual de madera. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (05 de Marzo de 2018). Requerimientos para el control de motores paso a paso. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (05 de 2018). Residuos de polvo en tallado manual. Quito, Pichincha.
- Pilatuña, Y., & Pillajo, I. (05 de Marzo de 2018). Sistema de control actual de la empresa. Quito, Pichincha, Ecuador.
- Pilatuña, Y., & Pillajo, I. (Mayo de 2018). Tallado de pieza de madera fresadora CNC. Quito.
- Saavedra, M. L. (2015). *Fresa Cutters Basic* . Obtenido de <http://slideplayer.es/slide/10249627/>
- Smith, P. (2003). *CNC Programming Handbook*. New york: Industrial Press, Inc.
- Teruel, F. C. (2005). *Control Numérico y Programación*. Madrid: Ediciones tecnicas.
- The MathWorks. (07 de Septiembre de 2016). *Matlab*. Obtenido de <https://es.mathworks.com/products/matlab.html>

ANEXOS

ANEXO 1: Diseño Electrónico



Elaborado por: Pilatuña & Pillajo

ANEXO 2: Máquina con sistema de automatización actual.



Elaborado por: Pilatuña & Pillajo